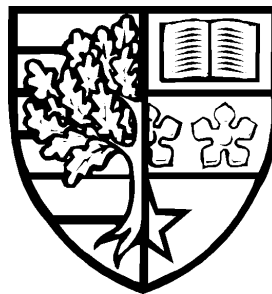


THE APPLICATION OF THREE-DIMENSIONAL MASS-SPRING STRUCTURES IN THE REAL- TIME SIMULATION OF SHEET MATERIALS FOR COMPUTER GENERATED IMAGERY

Bhopinder Singh Mahal



Submitted for the degree of Doctor of Philosophy

Heriot-Watt University

School of Engineering and Physical Sciences

October 2010

The copyright in this thesis is owned by the author. Any quotation from the thesis or use of any of the information contained in it must acknowledge this thesis as the source of the quotation or information.

ABSTRACT

Despite the resources devoted to computer graphics technology over the last 40 years, there is still a need to increase the realism with which flexible materials are simulated.

However, to date reported methods are restricted in their application by their use of two-dimensional structures and implicit integration methods that lend themselves to modelling cloth-like sheets but not stiffer, thicker materials in which bending moments play a significant role.

This thesis presents a real-time, computationally efficient environment for simulations of sheet materials. The approach described differs from other techniques principally through its novel use of multilayer sheet structures. In addition to more accurately modelling bending moment effects, it also allows the effects of increased temperature within the environment to be simulated. Limitations of this approach include the increased difficulties of calibrating a realistic and stable simulation compared to implicit based methods.

A series of experiments are conducted to establish the effectiveness of the technique, evaluating the suitability of different integration methods, sheet structures, and simulation parameters, before conducting a Human Computer Interaction (HCI) based evaluation to establish the effectiveness with which the technique can produce credible simulations. These results are also compared against a system that utilises an established method for sheet simulation and a hybrid solution that combines the use of 3D (i.e. multilayer) lattice structures with the recognised sheet simulation approach.

The results suggest that the use of a three-dimensional structure does provide a level of enhanced realism when simulating stiff laminar materials although the best overall results were achieved through the use of the hybrid model.

To Dad, we all miss you dearly.

ACKNOWLEDGEMENTS

I would like to thank my supervisors, Professor Jonathan Corney, Dr Doug Clark and Professor John Simmons for their trust and sustained encouragement throughout the project. Professor Corney deserves special praise for volunteering so much time and effort in guiding me through the final (and arguably hardest) stages of this work.

My deepest gratitude goes out to my parents, Gurdial Singh Mahal and Mohinder Kaur Mahal, and sister, Ranbir Mahal, who always supported my decision to undertake this work. Sadly, my father didn't have the opportunity to see me complete this work but the drive and commitment he gave me ensured I never gave up, even during the darkest of days.

I would like to acknowledge the generous support I received from friends and colleagues in the Department of Mechanical Engineering at Heriot-Watt University who have challenged my ideas and helped me to overcome problems. In particular I would like to thank Mr Ewan Brown, Dr Grant Fraser, Mr Ross Brown and Dr Jonathan Shephard for their dedication, time and patience. My thanks also go out to the volunteers who kindly participated in the HCI study.

Finally, to my wife Dawn, you helped me through the difficult times, giving me endless reassurance, support and understanding for more years than I deserved. You never questioned my motives for completing this project and have always remained my best friend and closest ally. This achievement would mean nothing without you. Thank you.

ACADEMIC REGISTRY

Research Thesis Submission



Name:	Bhopinder Singh Mahal		
School/PGI:	Engineering and Physical Sciences		
Version: <i>(i.e. First, Resubmission, Final)</i>	Final	Degree Sought (Award and Subject area)	Ph.D. in Mechanical Engineering

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

- 1) the thesis embodies the results of my own work and has been composed by myself
- 2) where appropriate, I have made acknowledgement of the work of others and have made reference to work carried out in collaboration with other persons
- 3) the thesis is the correct version of the thesis for submission and is the same version as any electronic versions submitted*.
- 4) my thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
- 5) I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.

* Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.

Signature of Candidate:		Date:	15.10.2010
-------------------------	--	-------	------------

Submission

Submitted By <i>(name in capitals)</i> :	BHOPINDER MAHAL
Signature of Individual Submitting:	
Date Submitted:	

For Completion in Academic Registry

Received in the Academic Registry by <i>(name in capitals)</i> :			
Method of Submission <i>(Handed in to Academic Registry; posted through internal/external mail):</i>			
E-thesis Submitted (mandatory for final theses from January 2009)			
Signature:		Date:	

TABLE OF CONTENTS

1	INTRODUCTION	1
1.1	THE ENGINEERING SCIENCE OF THIN LAMINAE	4
1.2	FLEXIBLE SURFACE MODELLING IN CGI	9
1.3	THE CHALLENGES OF LAMINAR MODELLING	12
1.4	RESEARCH HYPOTHESIS AND OBJECTIVES	13
1.4.1	<i>Objectives</i>	13
1.4.2	<i>Scope of the Thesis</i>	14
1.5	THESIS OUTLINE	15
2	FLEXIBLE SURFACE MODELLING – A REVIEW OF RELEVANT LITERATURE	16
2.1	INTRODUCTION	16
2.2	COMPUTATIONAL APPROACHES TO MODELLING MOVEMENT	18
2.2.1	<i>Rigid Body Dynamics and Physical Simulation</i>	19
2.2.2	<i>Finite Element Modelling</i>	21
2.2.3	<i>Particle Systems</i>	24
2.2.4	<i>Mass-Spring Systems</i>	26
2.2.5	<i>Curves and Surfaces</i>	29
2.3	SHEET SIMULATION AND DEFORMATION	34
2.3.1	<i>Simulation Dynamics - Modelling Ordinary Differential Equations</i>	36
2.3.2	<i>Finite Element Analysis and Finite Element Modelling</i>	39
2.3.3	<i>Cloth Based Particle Systems</i>	41
2.3.4	<i>Mass-Spring Sheet Simulations</i>	41
2.3.5	<i>Geometric Sheet Modelling</i>	47
2.4	THE RELATIONSHIP BETWEEN ANIMATION & SIMULATION	49
2.5	APPLICATIONS AND IMPLEMENTATIONS	52
2.5.1	<i>Cloth Simulation and Deformation</i>	52
2.5.2	<i>The Kawabata System's Fabric Material Properties</i>	58
2.5.3	<i>Skin Deformation and Virtual Actors</i>	61
2.6	GRAPHICS PROCESSING UNITS	63
2.7	DISCUSSION	65
2.8	SUMMARY	70
3	ASSESSING THE THEORY	71
3.1	INTRODUCTION	71
3.2	EVALUATION OF SHEET GENERATION FROM REAL WORLD PARAMETERS	73
3.3	VALIDATION OF THE SIMULATED MATERIAL	75

3.3.1	<i>Overview</i>	75
3.3.2	<i>Hooke's Law</i>	75
3.3.3	<i>Young's Modulus of Elasticity</i>	76
3.3.4	<i>Validation of the Sheet Modulus</i>	79
3.4	BENDING MOMENT REPRESENTATION	81
3.5	THE AURORA MODEL DYNAMICS.....	84
3.5.1	<i>Sheet Calculation Engines</i>	85
3.6	PHYSICAL SIMULATION	92
3.6.1	<i>Damping Model</i>	92
3.6.2	<i>Gravity</i>	97
3.6.3	<i>Simulation Instability</i>	97
3.7	PERIODIC MOTION	100
3.7.1	<i>Vibration</i>	100
3.7.2	<i>Harmonic Oscillation</i>	101
3.8	SUMMARY	102
4	SYSTEM IMPLEMENTATION	103
4.1	INTRODUCTION	103
4.2	SYSTEM DESIGN	104
4.2.1	<i>Design Space Analysis</i>	104
4.2.2	<i>Problems and Questions</i>	104
4.2.3	<i>Questions within the Design Space</i>	105
4.2.4	<i>Questions Options Criteria Diagram</i>	106
4.2.5	<i>Hierarchical Design Analysis</i>	107
4.2.6	<i>Unified Meta language (UML)</i>	107
4.3	THE SYSTEM SOFTWARE.....	113
4.3.1	<i>The Aurora System</i>	113
4.3.2	<i>Class Structure</i>	114
4.3.3	<i>Sheet Attributes</i>	116
4.3.4	<i>Sheet Relationships</i>	117
4.3.5	<i>Lightning Project Structure</i>	118
4.4	THE SHEETS.....	123
4.4.1	<i>The Sheet Data File Structure</i>	123
4.4.2	<i>Sheet Matrix Structures</i>	124
4.5	POLYHEDRA.....	129
4.5.1	<i>Data File Structure</i>	129
4.6	COLLISION DETECTION.....	131
4.6.1	<i>Overview</i>	131
4.6.2	<i>Sector Based Testing</i>	133
4.6.3	<i>Geometric Collision Testing</i>	134
4.7	SHADING THE SHEET	135
4.7.1	<i>Gourauld Shading</i>	135
4.7.2	<i>Phong Shading</i>	135
4.7.3	<i>Selection of the Shading Approach</i>	136
4.8	THE WORLD ON WINDOW VIRTUAL ENVIRONMENT	137

4.8.1	<i>The World on Window Interface</i>	138
4.8.2	<i>The Viewing Perspective</i>	140
4.8.3	<i>The Menus</i>	144
4.9	REAL WORLD TO SIMULATION CORRELATION	147
4.10	AURORA APPLICATION LIBRARY STRUCTURE	149
4.10.1	<i>Overview</i>	149
4.10.2	<i>Library Components</i>	150
4.11	THERMAL SOFTENING MODULUS	151
4.11.1	<i>Animation of Virtual Heat</i>	151
4.12	SUMMARY	157
5	EXPERIMENTAL TESTING.....	158
5.1	INTRODUCTION	158
5.2	REAL WORLD EXPERIMENTATION.....	159
5.2.1	<i>The Polyhedron</i>	159
5.2.2	<i>The Sheets</i>	159
5.2.3	<i>The Room Temperature Experiments</i>	160
5.2.4	<i>The Heat-Based Experimental Work</i>	164
5.2.5	<i>Simulation-Based Experimental Work</i>	168
5.3	AURORA SYSTEM TESTING	170
5.3.1	<i>Test Plan</i>	170
5.3.2	<i>Simulation Test Equipment</i>	170
5.3.3	<i>Test Schedule</i>	171
5.3.4	<i>Results of Sheet Dynamics Testing</i>	172
5.3.5	<i>Results of MaSSE Engine Performance Testing</i>	195
5.4	SUMMARY OF RESULTS.....	204
6	HUMAN COMPUTER INTERACTION EXPERIMENTATION.....	206
6.1	INTRODUCTION	206
6.2	THE SYSTEM BENCH MARK.....	207
6.3	HCI EXPERIMENTAL TESTING	210
6.3.1	<i>The Evaluation Test</i>	210
6.3.2	<i>The Questionnaires</i>	213
6.3.3	<i>The Test Population</i>	214
6.4	HCI TEST RESULTS	215
6.5	SUMMARY OF TEST RESULTS.....	219
7	DISCUSSION	220
7.1	INTRODUCTION	220
7.2	FULFILMENT OF AIMS	222
7.3	FURTHER SYSTEM EXPANSION	224
7.4	RECOMMENDATIONS FOR FUTURE WORK.....	226

7.5	CONCLUSIONS.....	229
	REFERENCES.....	230
8	APPENDICES.....	243
8.1	APPENDIX A - APPLICATION DEVELOPMENT	243
8.1.1	<i>Tool Selection</i>	243
8.1.2	<i>Hardware Selection</i>	243
8.1.3	<i>Development Environment</i>	244
8.2	APPENDIX B – PRINCIPLES OF PHYSICAL SIMULATION	247
8.2.1	<i>Mass Systems</i>	247
8.2.2	<i>Elastic Collisions in Physical Simulation</i>	248
8.3	APPENDIX C – HCI TESTING.....	252
8.3.1	<i>Evaluation Procedure Sheet</i>	252
8.3.2	<i>Background Questionnaire</i>	254
8.3.3	<i>User Satisfaction Questionnaire Part 1</i>	255
8.3.4	<i>User Satisfaction Questionnaire Part 2</i>	257
8.4	APPENDIX D – HCI EXPERIMENTAL DATA	258

FIGURES

Figure 1.1a – Reproducing Synthetic Table Cloth Behaviour (Escalight)	9
Figure 1.1b – Marilyn Monroe in ‘Flashback’ (Magnenat-Thalmann and Volino, 2005)	9
Figure 1.1c – Virtual Facial Skin Animation (Miralab EPOCH project)	9
Figure 1.2a – Genuine Table-Cloth Behaviour	10
Figure 1.2b – The ‘Real’ Marilyn Monroe in ‘The seven year itch’	10
Figure 1.2c – Real Facial Movement (demonstrated by Sir Ben Kingsley)	10
Figure 1.3a – Effect upon soft material with minimal stiffness	12
Figure 1.3b – Effect upon rigid material with significant bending stiffness	12
Figure 2.1 - Geometric Representations of Deformable Surfaces (Montagnet et al., 2001)	17
Figure 2.2a – The ‘Genesis Planet Explosion’ in Star Trek 3.	24
Figure 2.2b – Basic Particle Simulation	24
Figure 2.2c – Mud and Smoke Particles in Gears of War 2	24
Figure 2.3 - Mass Spring Deformations (Volino et al., 2007)	27
Figure 2.4 - Two Masses interconnected via a Spring	28
Figure 2.5 - Continuity of Piecewise Curves (Angel, 1996)	30
Figure 2.6 - A Mesh of Particles	42
Figure 2.7 – A simple n-degree of freedom mass-spring system (Rao, 1990)	43
Figure 2.8 – Forces placed upon Mass M_i	44
Figure 2.9 - Provot’s ‘Flexion’ System	54
Figure 2.10 – The Thalmanns Humanoid Cloth System	55
Figure 2.11 - Rotational stiffness and MaSSE bending model	67
Figure 3.1 – Overview of Sheet Generation from Real-World Properties	74
Figure 3.2 – Young’s Modulus Structure	77
Figure 3.3 - Rotational Stiffness Bending Model	81
Figure 3.4 - MaSSE Bending Model	82
Figure 3.5- MaSSE Release 1 Dynamics Flowchart Activity Diagram	86
Figure 3.6- MaSSE Release 2 Dynamics Flowchart Activity Diagram	88
Figure 3.7- MaSSE Release 3 Dynamics Flowchart Activity Diagram	90
Figure 3.8 – Time-Strain Curve for Spring	93
Figure 3.9 – Time-Strain Curve for Dashpot Damper	93
Figure 3.10 – Time-Strain Curve for Spring and Damper (Maxwell Element)	94
Figure 3.11 – Time-Strain Curve for Parallel Spring and Dashpot (Voigt-Kelvin Element)	94
Figure 3.12 - Simple Damping Model	95
Figure 3.13 – Strain curves for the Advanced Damping System (Higgins, 1994)	96
Figure 3.14 - Advanced Damping Model	96
Figure 3.15 – A Demonstration of Sheet Instability	98
Figure 4.1 – Pre-Simulation QOC Diagram	106
Figure 4.2 – Hierarchical Task Analysis Diagram	107
Figure 4.3 – Simulation Class Diagram	108
Figure 4.4 – Aurora Interface State Chart	110
Figure 4.5a – MaSSE Euler Activity Diagram	111
Figure 4.5b – MaSSE Euler-Cromer Activity Diagram	111
Figure 4.5c – MaSSE Euler-Richardson Activity Diagram	112
Figure 4.6 – Attributes of Mass and Spring Classes	117
Figure 4.7 – Mesh of SaMM Particles	120
Figure 4.8 – Mesh of SLaMM Particles	121

Figure 4.9 – Mesh of TriMM Particles	122
Figure 4.10 – Mass-Spring System Layout.....	125
Figure 4.11 - Multi-Layer Square (SaMM) Sheet Structure.....	126
Figure 4.12 – Multi-Layer Square Lattice (SLaMM) Structure	127
Figure 4.13 – Multi-Layer Triangular (TRiMM) Sheet Structure	128
Figure 4.14 – Two Dimensional Sector-Based Collision Detection.....	133
Figure 4.15 – The WoW interface structure	138
Figure 4.16 – The WoW interface.....	139
Figure 4.17 – Rotation about the scene.....	141
Figure 4.18 - Exploiting the Pan facility.....	142
Figure 4.19 – Zooming in and out of the environment	143
Figure 4.20 - Viewing a SlaMM sheet fall from an elevated left-side position.....	144
Figure 4.21 – The First Level Interface Menu	145
Figure 4.22 – The Aurora Library Structure	149
Figure 4.23 - Radiation Emission Diagram	153
Figure 4.24 – Demonstration of the off_centre_spg_weakening coefficient.....	156
Figure 5.1a – Triangular Wedge	159
Figure 5.1b – Partial Cube	159
Figure 5.2 – A 0.25mm Styrene Sheet Falling Onto the Triangular Wedge.....	160
Figure 5.3 – A 0.5mm Styrene Sheet Falling Onto the Triangular Wedge.....	161
Figure 5.4 – An Industrial Grade Sheet Falling Onto the Triangular Wedge.....	162
Figure 5.5 – A 0.4mm PVC Sheet Falling Onto the Triangular Wedge	162
Figure 5.6 – A 0.5mm Styrene Sheet Falling Onto the Partial Cube	163
Figure 5.7 – The Heat Experiment Testing.....	164
Figure 5.8 – A 0.25mm Styrene Sheet Melting over the Triangular Wedge	166
Figure 5.9 – An Industrial Grade Sheet Melting Over the Triangular Wedge	167
Figure 5.10 – Comparison of adapting Restitution Coefficient of sheet falling onto a flat surface	173
Figure 5.11a – Comparison of adapting Kinetic friction of sheet falling onto a sloping surface	174
Figure 5.11b – Comparison of adapting Kinetic friction of sheet falling onto a sloping surface	175
Figure 5.12 – Demonstration of a ‘bottom layer’ sheet falling onto complex shapes ..	175
Figure 5.13 – Demonstration of a ‘top layer’ sheet falling onto complex shapes	176
Figure 5.14 – Sheet Simulation comparison of NURBS functionality	178
Figure 5.15 – A Comparison of Spring Stiffness Variation Upon Sheet Behaviour	181
Figure 5.16 – A SaMM sheet falling upon a Triangular Wedge shape	182
Figure 5.17 – A SLaMM sheet falling upon a Triangular Wedge shape	183
Figure 5.18 – A TRiMM sheet falling upon a Triangular Wedge shape	184
Figure 5.19 – Comparison of stiff and Real world Sheet Behaviour over a Cube.....	186
Figure 5.20 – Comparison of Stiff and Real world Sheet Behaviour over a Wedge	187
Figure 5.21 – A Stiff Sheet Falling between Two walled Surfaces.....	188
Figure 5.22 – Simulation of Sheet folding over a Semi-Spherical Shape.....	189
Figure 5.23 – A SLaMM sheet falling and melting over a Semi-Spherical Shape.....	190
Figure 5.24 – A Industrial Grade Sheet Melting Over the Triangular Wedge	191
Figure 5.25 – Directional Virtual Heat on SaMM Sheet Deforming over a Wedge.....	192
Figure 5.26 – Directional Virtual Heat on SaMM Sheet Deforming over a Hemi-Sphere	193
Figure 5.27 - Sheet Size testing Scenario	198
Figure 5.28 - Simulation Result Timings for 4 Varying Sheet Sizes.....	200
Figure 5.29 - Simulation Timing Results for MaSSE Engines with Fixed Time-step..	201

Figure 5.30 - Timing Results for MaSSE Engines with Variable Time-step	202
Figure 6.1 – Structure of the 3D Hybrid Provot Sheet.....	208
Figure 6.2 – Flexion Rotational Stiffness Model	209
Figure 6.2 – First Evaluation: Styrene Sheet Falling onto a Cube.....	212
Figure 6.3 – Second Evaluation: Industrial Grade Plastic Sheet Falling onto a Wedge	212
Figure 6.4 – Third Evaluation: Styrene Sheet Falling onto a Wedge	213
Figure 6.5a – Cube Mean Values	215
Figure 6.5b –Wedge#1 Mean Values.....	215
Figure 6.5c –Wedge#2 Mean Values	216
Figure 6.6a –Cube Median Values.....	216
Figure 6.6b – Wedge#1 Median Values.....	216
Figure 6.6c - Wedge#2 Median Values	217
Figure 6.7a – Cube Standard Dev.	217
Figure 6.7b – Wedge#1 Standard Dev.	217
Figure 6.7c – Wedge#2 Standard Dev.	218
Figure 6.8a – Overall Mean Results.....	218
Figure 6.8b – Overall Standard Deviation	218
Figure 7.1 – Virtual Heat Source Simulation Approach.....	228
Figure B.1 - A Single Particle	247
Figure B.2 – Collision Detection with a Plane.....	249
Figure B.3 – Collision Response with a Plane.....	251

TABLES

Table 2.1 – Example Mechanical Properties of Wool-Polyester Cloth	59
Table 2.2 – Common Value Ranges for Fabric Materials	60
Table 2.3 – Suitability of Deformation Methods to Hypothesis Objectives.....	65
Table 3.1 – Sheet Values For SLaMM HCI Test Sheets	79
Table 3.2 – Typical Material Values For Young’s Modulus	80
Table 4.1 – Environment Dimensions.....	123
Table 4.2 – Real World Simulation Testing Parameters.....	147
Table 4.3 - MaSSE Engine Testing Variables	148
Table 5.1 – Simulation Testing Parameters	171
Table 5.2 - Sheet Dimensions	197
Table 5.3 - MaSSE Engine Testing Variables	198
Table 5.4 - Typical sheet parameters for MaSSE Engines	205
Table A.1 - Overview of Direct3D/DirectX and OpenGL.....	245

NOMENCLATURE AND SYMBOLS

Young's modulus	E	Elastic modulus tensors for metric deformation	C_G
Poisson's coefficient	ν	Elastic modulus tensors for curvature deformation	C_B
Rigidity modulus	G	Tensile strain	ε_θ
Strain energy density	U	Tensile stress	σ_θ
Strain tensor	$\lambda_{\alpha\beta}$	Bending strain	κ_θ
Elastic modulus tensors	$C^{\alpha\beta\delta\gamma}$	Bending moment	τ_θ
Indices for principle axis	α, β, γ and δ	Spring stiffness	k, k_s
Aggregation of strain energies (metric and curvature fabric deformation)	Σ	Damping constant	$d; k_d$
Stress due to the metric deformation	σ	Vector Point from Mass B to Mass A	L
Stress due to curvature deformation	τ	Damping Force Inverted on mass A	F
Strain due to metric deformation	ε	Natural Frequency (radians/sec)	ω
Strain due to curvature deformation	κ	Normal of Plane	N
Tensile modulus (principle axis)	D_1, D_2	Acceleration component in x, y, z system of coordinates, N/m	A_x, A_y, A_z
Shear modulus	D_s	Force component in x, y, z system of coordinates, N/m	F_x, F_y, F_z
Poisson's ratio	ν_1, ν_2	Displacement component in x, y, z system of coordinates	$D_x, D_y, D_z; x, y, z$
Bending modulus	H_1, H_2	Velocity component in x, y, z system of coordinates, N/m	V_x, V_y, V_z
Twisting modulus	H_s		
Shear modulus	D_s		
Physical analogue quantities to Poisson's ratios	μ_1, μ_2		

Time	t	Abbreviations	
Mass	m	Computational Fluid Dynamics	CFD
Acceleration	a	Computer Generated Imagery	CGI
Force	f	Degree Of Freedom	DOF
Velocity	v	Finite Element Modelling	FEM
Gravity (force)	g	Free Form Deformation	FFD
Runge Kutta: Constant weighting of the coefficients	W_j	Graphics Library Utility Interface	GLUi
		Graphics Library Utility Toolkit	GLuT
Runge Kutta: Order of the method	r	Mass Spring Simulation Engine	MaSSE
Runge Kutta: Estimates of change in y evaluated at r locations in range h	K_j	Microsoft Foundation Classes	MFC
		Non-Uniform Rational B-Spline	
Runge Kutta: Locations	r		NURBS
Runge Kutta: Range	h		
Runge Kutta: Acceleration	$\dot{v}; \ddot{y}$	Open Graphics Library	OpenGL
Runge Kutta: Velocity	$\dot{y}; v$	Ordinary Differential Equation	ODE
Young's modulus: sheet thickness	h	Partial Differential Equation	PDE
Young's modulus: spg width (z axis)	w	Square Mesh Matrix	SaMM
Young's modulus: spg width (x axis)	L	Square Lattice Mesh Matrix	SLaMM
		Triangular Mesh Matrix	TriMM
		Virtual Environment	VE
		World on Window	WoW

LIST OF PUBLICATIONS

The following journal and conference publications are based on this research.

1. Mahal, B.S., Clark, D.E.R. and Simmons, J.E.L. (2004) *A Pseudo-Immersive Virtual Environment - A Framework for Modelling Sheet Deformation*. *Virtual Reality Journal*, Vol. 8 (1), pp 1-16.
2. Mahal, B.S., Clark, D.E.R., Simmons, J.E.L. and Shanks, G.C. (2004) *Mass-Spring Systems and Heat Simulation - A Framework for Virtual Manufacture*. Proceedings *Simulation Interoperability Standards Organisation (SISO) Fall 2004 Workshop*. Orlando, USA, 19-24 September 2004.
3. Mahal, B.S., Clark, D.E.R. and Simmons, J.E.L. (2001) *Geometric Modelling of Sheet Deformation within a Virtual Environment*. *International Journal of Virtual Reality (IJVR) Vol. 5 (1)*. Publication delayed until 2003 but recorded as 2001 by publisher. Abstract only in printed version; full paper, 17 pp on CD-ROM.
4. Mahal, B.S., Clark, D.E.R. and Simmons, J.E.L. (2001) *Mass-Spring Simulation of Deformation in Elastic Sheet Structures'*, *Presence*, Vol. 10 (3), pp 331-342.
5. Mahal, B.S., Clark, D.E.R. and Simmons, J.E.L. (1999) *Sheet Deformation Modelling within a Virtual Environment*. Proceedings *6th UK Virtual Reality Special Interest Group (VR-SIG) conference*. Salford, UK, 13-15 September 1999, pp 167 -176.
6. Mahal, B.S., Clark, D.E.R. and Simmons, J.E.L. (1999) *Investigation of Sheet Bending using Mass-Spring Systems within a Virtual Environment*. Proceedings *2nd International Conference/Workshop on Presence*. Colchester, UK, April 1999.
7. Mahal, B.S., Clark, D.E.R. and Simmons, J.E.L. (1996) *Software Advances in Virtual Environments*. Proceedings *Framework for Immersive Virtual Environments (FIVE) International Conference*. Pisa, 20 - 22 December 1996, pp 67-69.

1 Introduction

The aim of this work is to produce a computational model of “stiff” laminar entities that generate physically plausible simulations rather than precisely replicating the actual behaviour of a particular sheet material. A prospective application of the research is the modelling of flexible laminar materials in the games industry as they interact with external forces. Such “qualitative” simulations are required to support various interactive and animated computer graphics applications (see section 2.6).

The reproduction of flexible laminar objects is a challenging endeavour due to the need to incorporate the effects of coupled deflections arising from both statics and dynamics. Furthermore in the real world the material properties change with the environment so heat (both latent and directional) influences the behaviour of the laminar body.

Such laminar modelling problems fall into the domain of ‘classical’ Newtonian mechanics utilising principles developed by Newton, Euler and Laplace in the 17th and 18th centuries. Consequently any introduction to the study of laminar modelling must start with an understanding of the underlying mechanics before considering the modelling challenges.

Dynamics forms the area of mechanics that studies how objects evolve and *move in time* under the influence of forces and constraints. The mathematical study of motion, known as kinematics, requires the three essential quantities: *position* (represented by a 3D coordinate system), *velocity* vectors and *acceleration* vectors. Dynamics requires the consideration of *mass* in addition to *force* (that is used to generate acceleration upon an object), energy (from which force is derived) and momentum (which is a product of mass and velocity relating to the amount of material movement).

The fundamental laws of dynamics that govern a rigid objects movement are well documented (see for example Beer et al., 2007). However these laws alone do not determine the nature of any resultant deformation (e.g. the nature of folding or creasing of the laminar bodies). For this, material properties such as *stiffness* and *flexibility* in various directions are required. Furthermore, additional properties (such as coefficients for *elasticity*, *plasticity* and *viscosity*) may also be required to represent the deformation characteristics of the material in variable environmental conditions.

In Computer Generated Imagery (CGI), the goal of a mechanical simulation system is a virtual reproduction of the mechanical behaviour of a lamina interacting within its environment. At any given moment a CGI simulation must be able to generate the *position* and *velocity* of the deformed lamina.

The parameters describing deformable surfaces can be broken down into four main families that are used to describe their mechanical behaviour.

1. Elasticity: that characterises the internal forces resulting from a given geometrical deformation.
2. Viscosity: that includes dynamic internal forces that result from a given deformation.
3. Plasticity: that describes how the properties evolve according to the deformation history.
4. Resilience: that defines the limit at which the structure will eventually break.

The elastic properties will usually provide the greatest influence upon the behaviour of the laminar sheets (e.g. plastic, cloth, paper etc.) although this will be dependent upon the simulation scenario. The effects of the remaining parameters are usually considered to be insignificant in the majority of reported CGI simulations. This is because the forces upon a sheet are often so small and fast-changing that the effects of viscosity, plasticity and resilience become insignificant. The elastic characteristics will now be examined in greater depth with the remaining parameters being studied in Chapter 3.

Elastic effects are divided in several components, which include *Metric Elasticity*, that describes deformation along the surface plane, and *Curvature Elasticity*, which represents deformations that are orthogonal to the surface plane.

Metric Elasticity forms the most essential characteristic of sheet deformation and as such, has been widely studied. It is broken down in terms of stress-strain relations, and for linear elasticity, the principal laws relating the strain ϵ to the stress σ involve the following parameters:

- The Young modulus E , which summarises the material's reaction (in relation to its Elasticity and Hook's Law) along the deformation direction.

- Poisson's ratio coefficient ν , which characterises the material's reaction (in relation to Poisson's Law) orthogonal to the deformation reaction.
- The Rigidity modulus G (relative to the Simple Shear Law), which pertains to the oblique reactions.

The mechanical principles linking stress and strain are well documented in existing texts (Ryder, 1973; Volino and Magnetat-Thalmann, 2000) however they do not directly lend themselves to real-time computer based visual modelling due to the need (in principle) to compute instantaneous values at every point of a flexible moving surface. So to bridge the gap between established engineering calculations and techniques that lend themselves to computer based modelling, a number of methods have been developed to allow sheet deformation to be simulated in an aesthetically accurate form within a real-time environment. These will be discussed again in greater depth during the course of the thesis. The other observation is that many environmental conditions such as heat and humidity will affect all these parameters.

Before that, the solid mechanics that is used to describe the laminar deformation process must be described.

1.1 The Engineering Science of Thin Laminae

Working under the assumption that a lamina is anisotropic and (in warp and weft directions[†]) orthotropic, it is possible for the mechanical properties of laminae formed by, say, cloth material to be measured by the Kawabata Evaluation System (see section 2.5.2). Three basic assessments are performed to determine the values examining:

1. Tensile properties,
2. Shear properties, and
3. Bending properties.

The mechanical properties of laminar materials are governed by the strain energy that is accumulated due to deformation. The two principal forms of deformation that occur in sheet draping are:

1. Metric deformation due to in-plane forces and
2. Curvature deformation due to out-of-plane forces.

The strain energy density U can be described by

$$U = \frac{1}{2} \sum \lambda^{\alpha\beta} C^{\alpha\beta\delta\gamma} \lambda_{\delta\gamma} \quad (\text{Equation 1.1})$$

Where $\lambda^{\alpha\beta}$ is the strain tensor due to the deformation and $C^{\alpha\beta\delta\gamma}$ is the elastic modulus tensor; α, β, γ and δ are indices to denote the directions of the principal axes (with respect to tensile and bending modulus); \sum is the aggregation of the strain energies due to metric and curvature deformation.

The elastic modulus tensors $C^{\alpha\beta\delta\gamma}$ are symmetric tensors, hence

$$C^{\alpha\beta\delta\gamma} = C^{\alpha\beta\gamma\delta} = C^{\beta\alpha\delta\gamma} = C^{\delta\gamma\alpha\beta} \quad (\text{Equation 1.2})$$

When the warp and weft directions coincide with the principal coordinate system of the fabric, $C^{1112} = C^{1211} = C^{1222} = C^{2212} = 0$ and so only five components of the tensor C^{1111} , C^{1122} , C^{1212} , C^{2211} and C^{2222} are non-zero.

[†] Unlike artificial materials such as lycra, woven cloth material is not isotropic, the two orthogonal directions defined by the thread orientation (referred to as *warp* and *weft*) have specific Young's modulus and Poisson's coefficients, D_1, ν_1 and D_2, ν_2 respectively.

From Hooke's law,

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} = \frac{1}{1 - \nu_1 \nu_2} \begin{bmatrix} D_1 & \nu_2 D_1 & 0 \\ \nu_1 D_2 & D_2 & 0 \\ 0 & 0 & D_s(1 - \nu_1 \nu_2) \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{bmatrix} \quad (\text{Equation 1.3})$$

$$\begin{bmatrix} t_{11} \\ t_{22} \\ t_{12} \end{bmatrix} = \frac{1}{1 - \mu_1 \mu_2} \begin{bmatrix} H_1 & \mu_2 H_1 & 0 \\ \mu_1 H_2 & H_2 & 0 \\ 0 & 0 & H_s(1 - \mu_1 \mu_2) \end{bmatrix} \begin{bmatrix} k_{11} \\ k_{22} \\ 2k_{12} \end{bmatrix} \quad (\text{Equation 1.4})$$

Where

σ and t are the stresses due to the respective metric and curvature deformation;

ε and k are the strains due to respective metric and curvature deformation;

D_1 and D_2 are the tensile modulus of the principal axis;

D_s is the shear modulus;

H_1 and H_2 are the bending modulus in the principal (*warp* and *weft*) axes;

H_s are the twisting modulus;

ν_1, ν_2 are Poisson's ratio;

μ_1 and μ_2 are the physical quantities analogue to the Poisson's ratios.

It can be observed that:

$$\nu_1 D_2 = \nu_2 D_1 \quad (\text{Equation 1.5})$$

and

$$\mu_1 H_2 = \mu_2 H_1 \quad (\text{Equation 1.6})$$

Hence the material modulus tensors can be expressed in terms of a set of measurable quantities (see equations 1.7a and 1.7b) where the C_G and C_B are the elastic modulus tensors due to the respective metric and curvature deformations (Wu et al, 2003).

$$\left. \begin{aligned} C_G^{1111} &= \frac{D_1}{1-\nu_1\nu_2} \\ C_G^{1122} &= C_G^{2211} = \frac{\nu_2 D_1}{1-\nu_1\nu_2} \\ C_G^{2222} &= \frac{D_2}{1-\nu_1\nu_2} \\ C_G^{1212} &= D_s \end{aligned} \right\} \quad (\text{Equation 1.7a})$$

$$\left. \begin{aligned} C_B^{1111} &= \frac{H_1}{1-\mu_1\mu_2} \\ C_B^{1122} &= C_B^{2211} = \frac{\mu_2 H_1}{1-\mu_1\mu_2} \\ C_B^{2222} &= \frac{H_2}{1-\mu_1\mu_2} \\ C_B^{1212} &= H_s \end{aligned} \right\} \quad (\text{Equation 1.7b})$$

It is frequently observed that not all “real-world” forces are required for a realistic visual simulation. For example, when a traditional solid object such as a horizontal fixed beam is subjected to a downward vertical force, the negligible visible deformation it creates means that only the metric stresses and strains need be considered.

For established soft fabrics CGI based applications (Rodriguez-Navarro et al., 2005) curvature may be calculated in the absence of H_1 or H_2 where such an omission would not be noticeable, however on stiff laminar objects this deficiency would result in the failure of a visually realistic simulation.

Should a sheet undergo a stretching force that produces the angle θ with one of the principal axis directions (symbolised by D_θ), then the tensile strain, denoted by ε_θ , is related to the tensile stress σ_θ , as illustrated in equation 1.8.

$$D_{\theta} = \frac{\sigma_{\theta}}{\varepsilon_{\theta}} \quad (\text{Equation 1.8})$$

Bending strain and bending moment can be expressed as κ_{θ} and τ_{θ} respectively with H_{θ} denoting the bending modulus in the principal axis:

$$H_{\theta} = \frac{\tau_{\theta}}{\kappa_{\theta}} \quad (\text{Equation 1.9})$$

The stress tensor transformation laws state that:

$$\varepsilon_{\theta} = [\cos^2 \theta \sin^2 \theta \sin \theta \cos \theta] \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{bmatrix} \quad (\text{Equation 1.10})$$

Where ε_{11} , ε_{22} and $2\varepsilon_{12}$ represent the tensile strain due to metric deformation

$$\begin{bmatrix} \sigma^{11} \\ \sigma^{22} \\ \sigma^{12} \end{bmatrix} = \begin{bmatrix} \sigma^{\theta} \cos^2 \theta \\ \sigma^{\theta} \sin^2 \theta \\ -\sigma^{\theta} \cos \theta \sin \theta \end{bmatrix} \quad (\text{Equation 1.11})$$

and σ^{11} , σ^{22} and σ^{12} represent the tensile stress due to metric deformation

where

$$\kappa_{\theta} = [\cos^2 \theta \sin^2 \theta \sin \theta \cos \theta] \begin{bmatrix} \kappa_{11} \\ \kappa_{22} \\ 2\kappa_{12} \end{bmatrix} \quad (\text{Equation 1.12})$$

Where κ_{11} , κ_{22} and $2\kappa_{12}$ represent the tensile strain due to curvature deformation

$$\begin{bmatrix} \tau^{11} \\ \tau^{22} \\ \tau^{12} \end{bmatrix} = \begin{bmatrix} \tau^{\theta} \cos^2 \theta \\ \tau^{\theta} \sin^2 \theta \\ -\tau^{\theta} \cos \theta \sin \theta \end{bmatrix} \quad (\text{Equation 1.13})$$

and τ^{11} , τ^{22} and τ^{12} represent the tensile stress due to curvature deformation.

The roots of equations 1.8 through to 1.13 can be found in strengths of materials literature (for example Ryder, 1973).

Substituting equations 1.10 - 1.13 into equations 1.8 and 1.9 and combining with equations 1.3 and 1.4 will provide both the tensile and bending modulus in the chosen direction, as highlighted in equations 1.14 and 1.15.

$$\frac{1}{D_\theta} = \frac{1}{D_1} \cos^4 \theta + \left(\frac{1}{D_s} - \frac{\nu_1}{D_1} - \frac{\nu_2}{D_2} \right) \cos^2 \theta \sin^2 \theta + \frac{1}{D_2} \sin^4 \theta \quad (\text{Equation 1.14})$$

$$\frac{1}{H_\theta} = \frac{1}{H_1} \cos^4 \theta + \left(\frac{1}{H_s} - \frac{\mu_1}{H_1} - \frac{\mu_2}{H_2} \right) \cos^2 \theta \sin^2 \theta + \frac{1}{H_2} \sin^4 \theta \quad (\text{Equation 1.15})$$

Since the Poisson's ratios of bending (μ_1 and μ_2) are typically very small, their value can often be neglected in any calculations, thus taking the angle $\theta = 45^\circ$, then the Poisson's ratio and the twisting modulus can be written as in Equations 1.16 and 1.17.

$$\nu_1 = \frac{1}{2} D_1 + \left(\frac{1}{D_1} + \frac{1}{D_2} + \frac{1}{D_s} - \frac{4}{D_{45^\circ}} \right) \nu_2 = D_2 \frac{\nu_1}{D_1} \quad (\text{Equation 1.16})$$

$$H_s = \left(\frac{4}{H_{45^\circ}} - \frac{1}{H_1} - \frac{1}{H_2} \right)^{-1} \quad (\text{Equation 1.17})$$

This result is exploited in section 2.5.2 where the mechanical properties of modulus D_1 , D_2 , D_s , H_1 , H_2 and H_s are measured directly by the Kawabata Evaluation System.

These equations have demonstrated the relationship between stress and strain. Attempting to compute the values in a computer-based simulation is significantly more difficult.




Although Finite Element Analysis (FEA) simulations are capable of modelling the effects of laminar behaviour, their slowness prohibits its application in real-time simulations. This shortfall is compounded once the additional complexity of modelling the impact of environmental conditions (such as heat) upon the laminate are considered. In later chapters it will be shown that mass-spring systems offer both speed and the ability to take into account environmental parameters and the representation of bending moment forces (such as the values of H_1 and H_2 in equations 1.4, 1.7b and 1.17).

The next section reviews established options for modelling or simulation of physical objects with the characteristics described above.

1.2 Flexible Surface Modelling in CGI

Computer Generated Imagery (CGI) dates back to the mid 20th century when Ivan Sutherland created his famous “sketchpad” system (Sutherland, 1963). Today, CGI is widely employed across a host of industries and applications that include computer games, training simulations, special effects for television and film through to industrial applications such as product modelling, design, performance-measurement and simulation (for example vehicle crash testing).

Figures 1.1a, 1.1b and 1.1c illustrate some recent examples of deformation in cloth simulation, avatar reproduction and methods of virtual clothes design compared to their real-world equivalents (see figures 1.2a, 1.2b and 1.2c).

<i>Figure 1.1a – Reproducing Synthetic Table Cloth Behaviour (Escalight[†])</i>	<i>Figure 1.1b – Marilyn Monroe in ‘Flashback’ (Magenat-Thalmann and Volino, 2005)</i>	<i>Figure 1.1c – Virtual Facial Skin Animation (Miralab EPOCH project[‡])</i>
		

[†] Produced by Escalight using 3D Studio Max.

<http://www.escalight.com/tutorials/3dsmax-tutorials/realistic-table-cloth-cover-part-1.html>




[‡] Miralab is a division of the University of Geneva.

http://www.miralab.ch//index.php?option=com_media&task=picture&Itemid=88&ID=Facial%20Animation&id_picture=0

Although the results of CGI initially proved to be slow, cumbersome and prohibitively expensive, today they are capable of offering near life-like levels of immersion with vastly reduced costs that enable the technology to be widely employed in many industrial and household applications.

Computer based animation has "traditionally" employed two-dimensional models for modelling sheet materials in real-time applications. Indeed the simulations presented in figures 1.2a, 1.2b and 1.2c illustrate the results of applying two-dimensional models to reproduce aesthetically life-like simulations. Interestingly all the methods shown required extensive periods of pre-processing in order to create a simulation. This limitation ultimately prohibits their use in real-time interactive environments where the simulations would be expected to respond in real-time to external (user) interaction.

Comparisons between figures 1.1a, 1.1b, 1.2a and 1.2b show that the views produced by CGI simulations are not designed to produce wholly authentic life-like reproductions of material behaviour, but instead to simply create aesthetically credible behaviour. The use of two-dimensional meshes to generate the scenes in the images also highlights a significant shortcoming of the technique in the use of implicit integration methods to produce stable simulations.

<i>Figure 1.2a – Genuine Table-Cloth Behaviour</i>	<i>Figure 1.2b – The ‘Real’ Marilyn Monroe in ‘The seven year itch’</i>	<i>Figure 1.2c – Real Facial Movement (demonstrated by Sir Ben Kingsley)</i>
		

It is evident in the comparison between figure 1.1a and 1.2a that the behaviour of the material resting upon the table does not provide a truly aesthetic representation. As is the case with most CGI applications, the emphasis has been to produce a reproduction that 'appears' to be accurate.

Figure 1.1b and 1.2b reinforces this concept with the virtual Marilyn simulation providing a basic reproduction of the grate-blowing scene from the Seven Year Itch movie without attempting to precisely mimic the complex cloth and wind dynamics that take place in the real-world scene.

Figures 1.1c and 1.2c highlight the real-world facial movement that is possible provided sufficient processing capabilities and processing time allow.

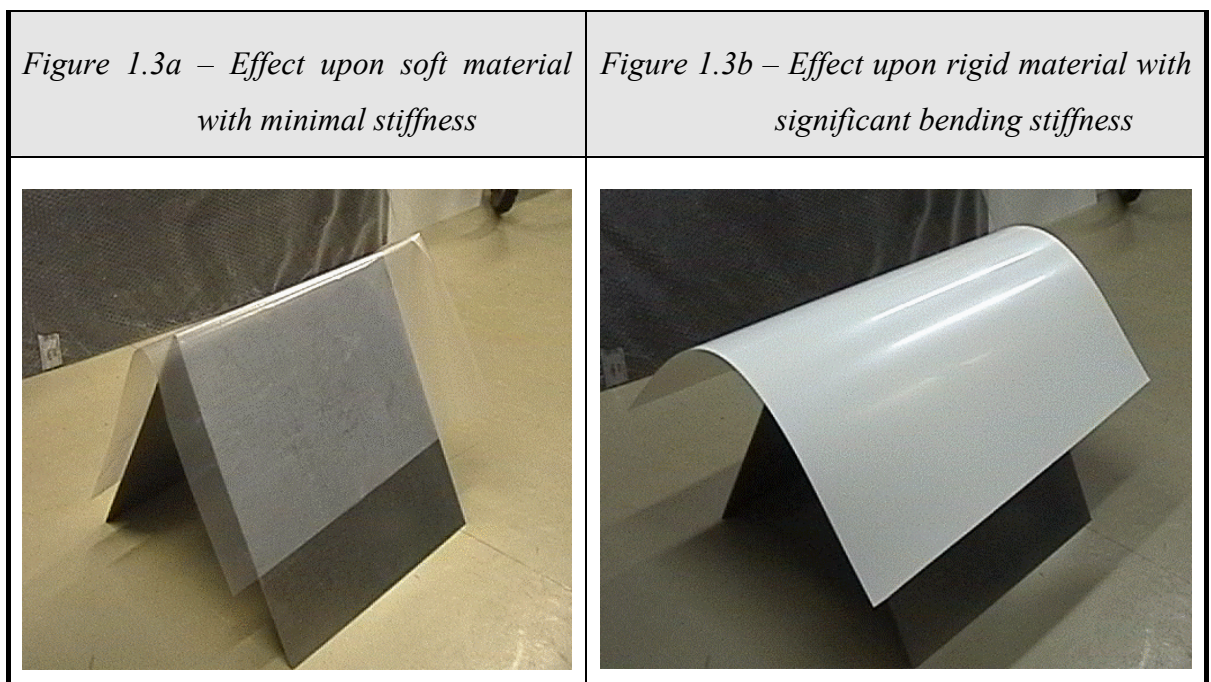
Research directions in deformation have covered a broad spectrum of techniques, with the main motivation being computer-based animation such as the modelling of cloth fabric (Magenat-Thalmann and Volino, 2005; Volino et al., 2005), soft substances (Desbrun and Gascuel, 1995), biomechanical modelling of human muscles (Maurel and Thalmann, 2000; Lewis et al., 2000; Aubel and Thalmann, 2000) as well as exploring how to burn (Liu et al., 2009) or bend (Kergosien et al., 1994) other virtual substances such as paper, sheet metal (Guan et al., 1997; Sun and Fiume, 1996) and polyethylene sheets (Sweeney et al., 2002). These methods all require advanced collision detection methods which have started to offer enhanced levels of accuracy with minimal computational overhead (Volino and Magenat-Thalmann, 2000; Volino and Magenat-Thalmann, 1994).

1.3 The Challenges of Laminar Modelling

As we will see in Chapter 2, The effects of bending moment are often neglected in simulations of laminar objects. However the effect of a bending moment upon a lamina has a significant impact upon the visual representation of the sheet not only as it interacts with other objects but also potentially as it comes to rest.

These differences are highlighted below as a sheet with minimal stiffness (figure 1.3a) and a sheet with a significant stiffness values (figure 1.3b) rest upon the top edge of a wedge shaped object.

As demonstrated in Chapter 5, what is needed to improve the realism of sheet modelling is a method of modelling the behaviour of laminar material which can support a significant bending moment (as demonstrated in figures 1.3a and 1.3b).



1.4 Research Hypothesis and Objectives

A principle for a model of any flexible material is that it should be capable of dynamic behaviour whilst being both numerically and computationally stable and efficient.

Based upon this statement, the underlying hypothesis driving this body of work is that:

“It is possible to create credible, real-time simulations of stiff laminae if bending moment are taken into account”

Implicit in this statement is the understanding that environmental conditions will be incorporated in the model.

1.4.1 Objectives

To support this hypothesis, the research must fulfil a number of set objectives, which are broken down below.

1. The construction of a computationally inexpensive environment that allows a model to be manipulated and deformed easily and with sufficient accuracy.
2. The development and evaluation of an approach for modelling sensitive sheet materials in real-time. This technique will utilise varying degrees of numerical complexity to consider the most suitable combination for a given scenario.
3. The production of a prototype to investigate the effectiveness of the proposed methodology.
4. To draw comparisons between the three-dimensional structures with (i) an existing two-dimensional sheet method, and (ii) real world testing to evaluate the effectiveness of the system. This will be achieved through the adoption of HCI based principles and an evaluation encompassing a limited population sample.
5. Explore the suitability of the system to modelling the influence of thermal deformation.

1.4.2 Scope of the Thesis

Sheet modelling research uses a breadth of methods, integration techniques and calculation methods, to reproduce the behaviour of materials ranging from cloth (Meng et al., 2010; Pabst et al., 2008; Rodriguez-Navarro and Susin, 2006) to sheet metal to human skin (Zhong et al., 2009; Boissieux et al., 2001). The thesis focuses upon the modelling of bending moment in laminar modelling and its relationship with the use of differing sheet structures, namely three-dimensional models compared to traditional two-dimensional laminar structures. The purpose of the experimental work was to clarify whether these new structures could produce a reasonably aesthetically accurate representation of sheet deformation. A limited study was conducted to assess the suitability of the mass-spring model to mimicking the aesthetic behaviour of laminae subject to thermal effect leading to loss of stiffness. The rationale of the HCI study was to provide an initial appraisal of the three-dimensional structure performance in relation to traditional two-dimensional meshes.

1.5 Thesis Outline

The introduction presents the inspiration behind the work, examining the basic principles of lamina deformation, the goals of this research, its underlying hypothesis and overall scope. The remaining chapters are summarised below.

Chapter 2 contextualises the research and offers a literature review of current advances in the fields of physically based modelling of laminar materials, dynamic simulation and object deformation, before examining the most widely researched theme of laminar CGI, that of modelling cloth.

Chapter 3 focuses upon the structure of the proposed system. It examines the use and behaviour of Mass-Spring systems and the MaSSE calculation engines, and sheet damping approaches.

Chapter 4 examines the design of the system, the structure of the three-dimensional sheet models and the World on Window Interface. The section also describes the use of shading and smoothing algorithms, collision detection, the correlation between simulation and real-world parameters, and how basic thermal effects were implemented in the system.

Chapter 5 reports on the real-world experimental tests that were conducted and draws direct comparisons with simulated examples.

Chapter 6 employs a user assessment to validate the work in Chapter 5.

Chapter 7 discusses the results of the work conducted in Chapters 5 and 6, drawing conclusions on the effectiveness of the approach, before considering areas of future work resulting from this thesis.

2 Flexible Surface Modelling – A Review of Relevant Literature

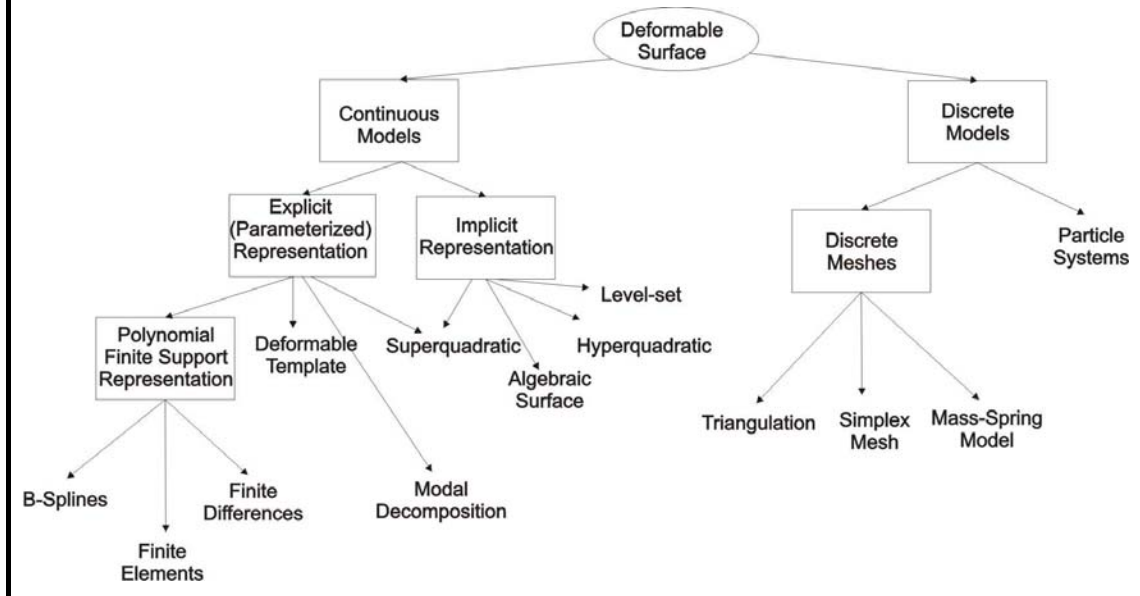
2.1 Introduction

This chapter reviews reported work that relates to the modelling of specific forms of deformable solids and laminae. In particular, following on from this, the work will examine current research in deformable solids, principally laminar materials, to establish the state of the art and enable the placement of this work within a broad context.

Montagnet et al. (2001) outlined the differing forms of deformable surfaces detailing the numerous forms of topology, geometry and deformation. He went on to describe how applications could not simply be classified as a specified method and that applications often used a hybrid of approaches to achieve their simulation.

Figure 2.4 is used to outline the different geometric representations of deformable surfaces.

Figure 2.1 - Geometric Representations of Deformable Surfaces (Montagnet et al., 2001)



The examination of the mechanical properties of cloth is intended to highlight the key areas of engineering that require understanding for any cloth based simulation to operate successfully.

This chapter will also introduce the tools necessary to realise the discussion later in this thesis.

2.2 Computational Approaches to Modelling Movement

Dynamic simulation and the mimicking of the behaviour of physical phenomena form one of the cornerstones of graphics applications. In its simplest form, through the utilisation of fundamental physical and mathematical laws, the behaviour of a simulated model can be reproduced to illustrate equivalent actions in the real world. This depicts the conduct of simple objects as they are acted upon by real world forces such as gravity or the effects of an action. One example would be throwing a ball across a room, which would be directly modelled using Newtonian physics. However, when attempting to reproduce complex reactions like deforming ‘soft’ objects such as cloth or the skin of a virtual human arm, then the represented bodies are typically broken down into simple engineering entities that combine to model behaviour (e.g. *masses* and *springs*).

As the cost of computational power continues to fall, the scope of computing simulation has continued to widen with the breadth of topics encompassed by physical modelling continuing to grow. New approaches have been devised for modelling new substances and complex dynamic behaviour that previously would have been considered too expensive to reproduce. Examples include granular materials such as sand (Onoue and Nishita, 2005), and hair on virtual animals and humans (Bando et al. 2003). The output of this research has already started to produce benefits for established fields such as traditional and computer-based animation (Nealen et al., 2006), computer-based games, special effects for television and film, Computer Generated Imagery (CGI) and computer-based graphics.

This diversity of applications has inevitably led to the formation of subdivisions, some of which are examined later. The topics selected will concentrate upon the principal components of the study, i.e. sheet based object deformation and the effect of heat-like thermal softening. Other disciplines covered include dynamic behaviour of objects which is often the cause of deformation, and collision detection. This expansion of simulation techniques for non-rigid based objects was examined by Cugini et al. (1999) who discussed the advantages and disadvantages of each approach. Some of these divisions are examined below.

Computer Simulation (and in its broader sense, animation) divides naturally into a mix of categories that are used to describe (i) the type and (ii) the nature of animated

objects, as well as (iii) the programming techniques that are used to achieve the animation.

Before discussing sheet simulation in depth we shall briefly review the four major methods currently employed in animation that include:

1. Rigid Body Dynamics
2. Finite Element Modelling
3. Particle System
4. Mass-Spring Method
5. Curves and Surfaces

After introducing each of these methods, their specific application to sheet modelling is discussed in greater detail. The scope of research presented impacts upon most of these categories to a varying degree.

Having reviewed each category then sections 2.3 and 2.5 will focus upon how these animation methods may be applied for the purpose of sheet simulation.

2.2.1 Rigid Body Dynamics and Physical Simulation

Rigid Body Dynamics has been widely adopted in the entertainment industry where these toolkits (such as PhysX and 3D Studio Max) and development environments (such as OpenCL/OpenGL and Microsoft's DirectCompute) are now regularly used in the games, special effects and computer-based animation film production process. Examples of physical simulation include modelling the falling of objects under the effects of gravity or collisions between foreign bodies (for example, collisions between snooker balls). As its name suggests, physical simulation is used to represent the physical behaviour any rigid object and as such covers a breadth of topics (see Appendix B for examples).

Filipcuk and Nikiel (2008) have presented a real-time, PC based boat simulation that utilises a mass-spring particle system to mimic the behaviour and forces generated by a sail as it drives a sail boat through the water. The model considers the effects of wind

and gravity upon the sail shape, although there is no rupture capability within their existing model. As such the shape of the sail will vary based upon its angle of incident and wind force. As reported the work only utilises a basic Euler integration method for force calculations, with minimal collision detection algorithms relating to the behaviour of the hull, hydrodynamic behaviour, and the boat's interaction with waves etc. The aerodynamic behaviour of the model relates to the sail as it drives the boat whilst simultaneously encountering the hydrodynamic forces resulting in friction upon the boat's movement. The sail is represented as an elastic body with the virtual wind creating an aerodynamic force upon it that forces it to become straightened, while gravitational forces attempt to pull it downwards. Through the use of varying spring values, different sheet materials can be modelled. The mainsail is ultimately limited to 120 faces, 45 vertices that are attached to 45 mass-spring particles and possesses only 1 degree of freedom (DOF).

Dynamic behaviour has formed the cornerstone of research by INRIA and iMAGIS, a joint project between INRIA, CNRS, Université Joseph Fourier and Institut National Polytechnique de Grenoble (Debonne et al., 2000). Although initially conducting research into recreating dynamic, life-like behaviour of objects, their approaches have been extended to include other problems attributed to physical simulation, which includes modelling the interaction and deformation of soft, highly plastic substances (Desbrun and Gascuel, 1995; Desbrun et al., 1996); advanced collision detection algorithms (Joukhader et al., 1997); the creation of autonomous virtual avatars and actors; forward and inverse kinematics; and immersive Computer Generated Imagery software.

In the field of dynamic simulation, Joukhader et al. (1997) have examined and drawn comparisons with two principal forms of dynamic contact, the 'penalty' and 'impulse' based models. The penalty model is based on the continuous representation of the collision phenomenon at a macroscopic level measuring the deformation of colliding bodies. However the impulse model relies upon the assumption that objects colliding at high speed undergo instantaneous, impulse based reactive forces meaning that it is not necessary to model what actually occurs during the process of collision. The investigation indicated that the penalty method can (with some optimisation) be an effective approach to simulating mass-spring models. This is contrary to thinking by

Baraff (1992) and Mirtich (1996) who both claimed that impulse-based models provided the only realistic alternative to rigid body simulation.

Lamouret and Gascuel (1996) proposed a “scripting” approach to modelling the dynamic behaviour of objects for use by animators who simply plot a trajectory and arrange the time-key framing and synchronisation with other objects (thus catering for a high degree of artistic freedom) with the software dealing with collision detection and response utilising Gascuel’s collision detection algorithm (Gascuel, 1993).

Later work conducted by iMAGIS with Caltech pushed forward the current technology in producing deformable, fast cloth modelling algorithms (Meyer et al., 2000), as well as soft and highly plastic substances (Debunne et al., 1999; Debunne et al, 2000).

Kim, and Venrovsky (2008) have devised a cause and effect based system whereby the variation of a model (e.g. a character’s pose) drives the behaviour and shape of a secondary (or driven) model, for example the clothes a virtual actor is wearing. Exploiting the correspondence between the two sets of shape deformations presented, with the use of pose shape deformation in real-time highlights the novelty of their approach to allow an animator to quickly produce results without waiting for significant levels of computation or time required prior to viewing the results of a scene. The authors have created a database of pre-calculated poses that relate to the behaviour of a particular garment upon an actor. This database is then utilised when creating an animation to model the behaviour of the clothing through determining which pose relates most closely to the actor’s behaviour, before extrapolating the stored pose shape onto the garment and placing it over the actor.

This use of secondary drivers is starting to gather interest from researchers, such as Zhong (2010).

2.2.2 Finite Element Modelling

The Finite Element Method is a numerical technique which gives approximate solutions to differential equations that model physics and engineering based problems. This is based upon the principle of discretisation, partitioning objects into smaller, manageable parts, by first breaking down a structure into distinct non-overlapping regions known as

elements (Chandrupatla and Belegundu, 1997). These component elements are then individually examined and each given values to represent stiffness.

Although a system (or object domain) can be discretised into smaller systems, components or finite elements, an important principle underlying the technique is that the final aim is to combine the understanding of each of the component elements to create a picture of the whole or continuous nature of the system itself.

According to Pereira and Moitinho de Almeida (1999), the finite element method is today established as the most important tool used by engineers for the analysis of complex structures. However, it is well known that the quality of the solutions obtained are closely related to the dimension of the mesh that is used and that “good” results will only be obtained when a large number of elements are used in areas encountering high-stress gradients. Chang (Chang et al., 2009) state that even with extensive computation time, Finite Element Analysis does not provide perfect results but instead provides ‘useful guidance’ for design.

The Finite Element Method is applicable to a wide range of engineering problems with several classifications for the problem types (Hinton and Owen, 1979). One approach is to define the engineering specialities to which the method may be applied, which include stress, thermal and geotechnical analysis problems. A further classification method involves the definition of the types of algebraic equations which result after discretisation by finite elements. The separate groups are:

- Equilibrium Problems - Where the system does not vary with time
- Eigenvalue Problems – These are extensions of Equilibrium problems where specific or critical values of certain parameters must be determined (e.g. stability and free vibration of linear elastic structures)
- Propagation Problems – Where transient phenomena take place (e.g. the dynamic analysis of elastic continua).

Modern day simulation applications incorporate animation techniques to visualise any real world physical behaviour (Nesme et al., 2010). The bases of these techniques are mathematical, physically-based modelling strategies where mass, energy, inertia,

velocity, acceleration, rotation, friction, temperature and gravity all have numerical representation.

Finite Element Modelling (FEM) and Computational Fluid Dynamics (CFD) represent two fundamental simulation techniques with FEM decomposing rigid objects as a lattice of nodes which can be subjected to forces and/or variations in temperature with the results being animated over time. CFD simulates the behaviour of gases and fluids as they interact with solid objects so that natural flow patterns and areas of turbulence can be identified.

Finite-Element Modelling (FEM) forms the most prevalent industrial approach to predicting the stresses and strains placed upon objects and their resulting behaviour. The technique has proven itself as robust and dependable with typical FEM applications that vary from predicting the locus of component part failures to calculating the maximum load of mechanical parts and eventual deformation behaviour of objects.

As a testament to its reliability, all major automotive and aerospace manufacturers utilise the technique extensively for evaluating the crash-worthiness of products, determining thermal stresses within engines and predicting failure rates of key components, possibly years prior to prototype construction.


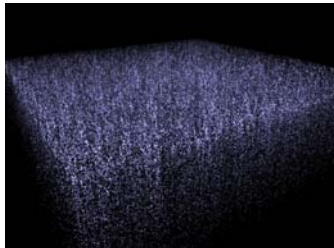

Thus, it is unsurprising that animators and special effects artists have extensively adopted these techniques to produce photo-realistic visual simulations. The drawback of FEM is its level of complexity and computational resource needed for even the simplest production (Chang et al., 2009). This results in a restrictive and non-interactive toolkit that discourages an animator from ‘playing’ with the model to produce the most aesthetically pleasing scene (Choi and Ko, 2005). However new developments such as GPU programming are starting to allow FEM to be modelled more regularly than before (Rodriguez-Navarro and Susin, 2006).

2.2.3 Particle Systems

A particle system is a collection of independent objects, often represented by a simple shape, dot (or mass particle). Each particle can be used to model many regular or irregular types of natural phenomena that include explosions, fire, smoke, sparks, waterfalls, clouds, fog, petals, grass and bubbles.

The basis of a mass particle is examined in Appendix B.

Examples of particle systems in films, simulations and computer games are demonstrated in figures 2.2a, 2.2b and 2.2c, highlighting the flexibility of the method.

<i>Figure 2.2a – The ‘Genesis Planet Explosion’ in Star Trek 3.</i>	<i>Figure 2.2b – Basic Particle Simulation.</i>	<i>Figure 2.2c – Mud and Smoke Particles in Gears of War 2.</i>
		

In a system, each particle will have its own set of properties related to its behaviour (for example, velocity, acceleration, etc.) as well as its look (for example, colour, shape, image, etc.). The mechanical behaviour of the material is implemented between the particles with interactions usually limited to the neighbouring particles for simulating the basic elasticity behaviours of the object.

Particle systems typically consist of three essential parts (depending upon the simulation), that without which the particle system will not function. These are:

1. The particle system itself
2. At least one particle, and
3. A particle emitter.

The basic components of a particle system will now be examined below.

Particles - A particle is the basis of a particle system (see appendix B). Particles are created from types like instances are created from objects; however, no collision detection is necessarily required (as this is dependent upon the simulation). Like objects, particles have properties that affect their behaviour and may also have other properties that define how they look and how they behave.

Emitters - A particle emitter creates particles. Emitters also have a region that defines where a particle appears. This can be a square, an ellipsis, a diamond shape, or even just a single point. Regions are used because particles are supposed to give the impression of being random, however in truth they can be controlled or restricted to any desired range.

Attractors - Particle Attractors attract particles.

Changers - Particle Changers change particles into another kind of particle.

Deflectors - Particle Deflections push particles away from them. Since the deflector itself is not visible, some nice effects can be achieved.

Destroyers - Particle Destroyers destroy particles as they come into contact with them.

Volino and Magnenat-Thalmann (2006) have demonstrated how particle based systems can be used to represent bending stiffness for implicit integration based cloth simulation. This is achieved by using a “bending vector” to represent the surface bending through a combination of linear particle positions before redistributing the vector as a particle force according to the surface bending stiffness.

Other applications that utilise particle systems include modelling the behaviour of soil when being manipulated by diggers (Li and Moshell, 1993) and to animate highly deformable bodies through the utilisation of smoothed particles (Desbrun and Gascuel, 1996) and other granular material (Onoue and Nishita, 2005; Luciani et al, 1995).

The relationship between particle systems and sheet deformation is examined further in section 2.3.3.

2.2.4 Mass-Spring Systems

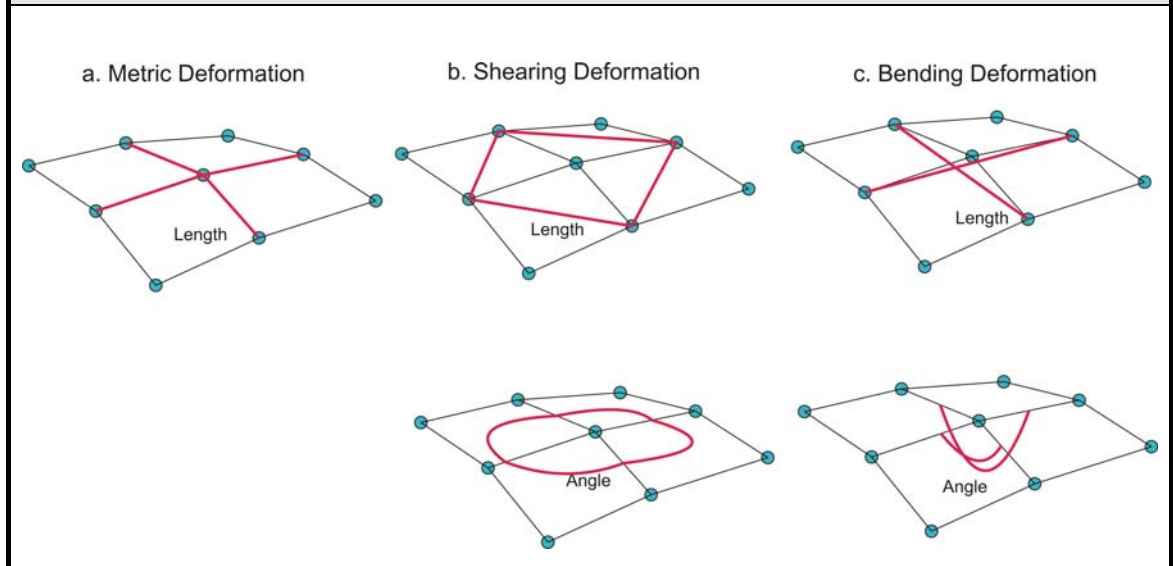
Mass-Spring systems form the simplest method of designing a volume model that utilises a particle system. In such arrangements, each particle represents a point-mass that forms a part of the material discretisation and links to its neighbouring masses through a “spring” structure that represents the elastic behaviour of the material and is used to also retain the particle’s initial resting positions.

Differing types of springs can be employed to represent the various mechanical properties of a laminar material as it undergoes deformation. The use of regular grid structures allow metric, shearing and bending elasticity to be modelled through the employment of spring elongation and so called “flexion springs”. Metric elasticity is usually defined through the elongation of springs that are placed along the lattice edges, shearing elasticity can be modelled by either the lattice angle springs or by diagonal elongation springs that operate obliquely across the grid structure.

Curvature elasticity may also be defined either through the use of flexion springs between opposing edges or through the elongation of springs between opposed vertices. Figure 2.3 illustrates the lengths or angles for measuring deformations in a square particle grid system.

When modelling cloth based systems using 2D structures, the use of elongation and flexion springs are not without their shortcomings. Elongation springs do not provide suitable reproductions in high deformation simulations, and although flexion springs are able to provide accurate levels of shearing representation, they are not only computationally expensive but (due to their computational cost) are also unsuitable for modelling surface orientations.

Figure 2.3 - Mass Spring Deformations (Volino et al., 2007)



Spring and Mass Systems are a highly flexible and adaptable method for modelling the dynamic behaviour of objects as they are subjected to loads and external forces (Volino et al., 2007; Wallner, 2007). To achieve this, the entity being modelled is often broken down into a series of mass points that are connected together through a series of spring elements that are used to transfer forces from a mass element to its neighbouring mass elements.

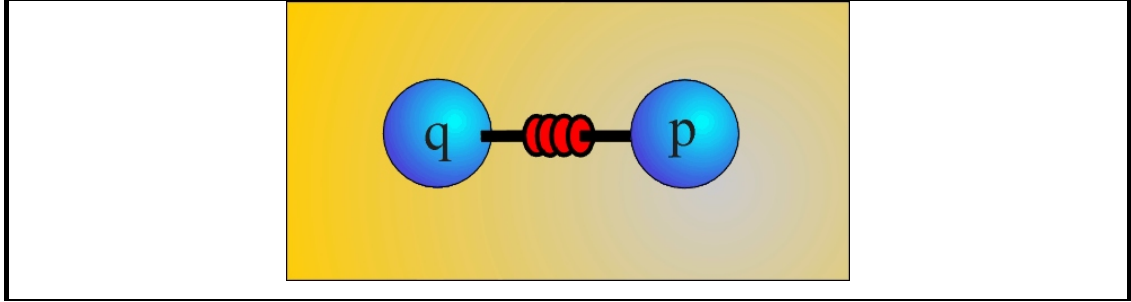
For a breakdown of calculations, consider two adjacent masses connected by a spring (see Figure 2.4), with:

f denoting the force acting upon particle p from particle q

s representing the spring's undisplaced, or resting length.

When stretched, the spring is subject to Hooke's law (see Equation (2.1)) and will generate a force that will act in the direction $d = p - q$, along the direct line between the points.

Figure 2.4 - Two Masses interconnected via a Spring



The symbol k_s represents the spring constant (or value for stiffness) and describes the situation as the particles move further apart, then the greater the force will be applied to bring the particles back together again.

$$\mathbf{f} = -k_s (|\mathbf{d}| - s) \frac{\mathbf{d}}{|\mathbf{d}|} \quad (\text{Equation 2.1})$$

When implemented in this simple form, the mass-spring system will start to oscillate and, if left undisturbed, will continue forever. To cater for this eventuality, a level of damping must be incorporated into Hooke's original equation (in the form of a friction coefficient). This results in the equation:

$$\mathbf{f} = - \left(k_s (|\mathbf{d}| - s) + k_d \frac{\dot{\mathbf{d}} \cdot \mathbf{d}}{|\mathbf{d}|} \right) \frac{\mathbf{d}}{|\mathbf{d}|} \quad (\text{Equation 2.2})$$

Where k_d is the damping constant, and

$$\dot{\mathbf{d}} = \dot{\mathbf{p}} - \dot{\mathbf{q}} = \mathbf{v}_p - \mathbf{v}_q \quad (\text{Equation 2.3})$$

The effect of this is to generate a proportional force that acts in the opposite direction to the spring force.

Likewise, when dealing with elastic materials, if two masses are forced together so that the distance between them is less than the undisplaced length of the spring joining them, then a repulsive force is generated.

At any one time-step, several parameters are invoked to describe the behaviour of each control point within free space. These are mapped using three-dimensional vectors.

Parameters include the mass's current location, the direction of any applied forces, the current *point* velocity and acceleration.

A value for the actual mass of the control point is also maintained. The *mass* instances are interconnected through a network of *spring systems* which also have properties that are used to describe the behaviour of the *spring* during the simulation.

Spring data includes information about the relative position of the *spring* within the sheet model (using pointers to its interconnected *masses*), the degree of axial spring *stiffness* and its undisplaced length. These three properties come together during the simulation to determine the behaviour of the object being modelled.

2.2.5 Curves and Surfaces

All graphics hardware at its most fundamental level produces points, line segments and polygons that are usually comprise of triangles and quadrilaterals. As the granularity of surfaces is reduced, the number of points and triangles required to produce a smooth flowing plane becomes prohibitive with respect to performance and efficiency. Thus it has now become common for three-dimensional computing applications to utilise curve and surface smoothing algorithms. One method around this problem is to utilise Bézier curves that use small numbers of control points to modify the behaviour of lines to form smooth and accurate curves. The work of Bézier was later extended to cater for complex curves and surfaces, eventually leading to the development of Non-Uniform Rational-B-Spline (or NURBS) curves and surfaces. Both of these concepts are examined in greater detail below.

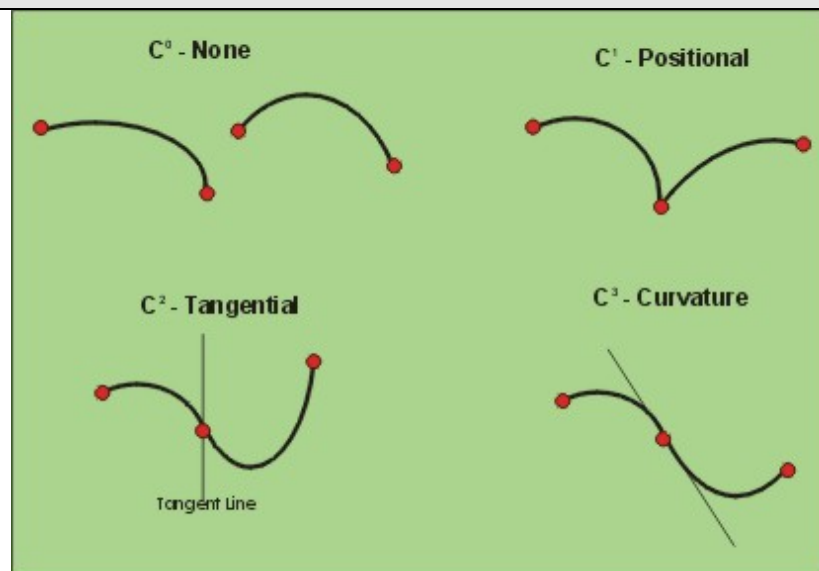
Bézier Curves

Bézier curves use control points and evaluators to direct the path and curvature of lines and surfaces. In Bézier curves the first and last control points actually form the start and end positions of the curve with the remaining control points acting as 'magnets' to manipulate the direction of the curve or surface. The *order* of the curve is defined as the number of control points, whilst the *degree* of the curve is one less than the value of

the *order*. The order represents the number of coefficients used to represent the parametric equation for the curve, whilst the degree corresponds to the highest exponent of the parametric parameter. A final attribute of Bézier curves is the use of continuity operators. These are used when two curves share an endpoint (known as a breakpoint) to combine and form a single piecewise curve. These represent the overall level of smoothness that is achieved where the curve merges with another. There are four categories of continuity operator, which are:

- C^0 (or None) where the curves do not meet.
- C^1 (or Positional) where the breakpoint simply acts as a join between the curves.
- C^2 (or Tangential) occurs when the two curves have the same tangent as the breakpoint.
- C^3 (or Curvature) means that the tangents of the two curves also have the same rate of change as the breakpoint, thus resulting in a smoother transition.

Figure 2.5 - Continuity of Piecewise Curves (Angel, 1996)



The differences between the categories are highlighted in figure 2.5. For complex curves and surfaces, one would strive to achieve C^2 or C^3 continuity.

The shortfalls of Bézier curves appear once the number of control points is increased, causing the smoothness of a curve to break down as the additional points pull on the curve.

Non-Uniform Rational-B-Spline (NURBS) Curves

One solution to the problem of Bézier curves is the use of bi-cubic-splines (or B-splines). These work in a similar manner to Bézier curves but with the added benefit that the curve is broken down into segments, with the shape of each segment being controlled by the nearest four control points. This results in each segment mimicking the behaviour of a fourth-order Bézier curve, with an overall curve that is smooth and benefiting from each segment exhibiting C^3 continuity. In addition, this means that the curve will not necessarily pass through any of the control points. The advantage of this is that a smoother curve is created, however the result of this smoothness may be at the cost of the curve's position accuracy.

The true benefits of NURBS is the ability to adapt the influence of the four control points for any given curve segment to enhance the smoothness of the curve, with this control being manipulated through a sequence of values known as *knots*. Each control point contains two non-descending knot values, the range of the knot values match the u and v parametric domain of a surface.

Cheng and Topping (1999) explain the suitability of NURBS for parallel mesh generalization, once the original domain is decomposed into several subdomains and the original boundary curve is subdivided into several individual NURBS segments.

Over the years, significant numbers of researchers have adopted the use of NURBS surfaces to strengthen the visual effects of their applications. Preston and Hewitt (1994) and Lamousin and Waggenspack (1994) both describe early examples of using NURBS to enhance free-form object deformation in animation.

Due to their effectiveness in providing effective, controllable and computationally inexpensive results, NURBS have been used extensively in modelling clothes, outer skin and muscle surrounding skeletal structures, curved objects such as internal body organs. One of the difficulties of utilising NURBS surfaces are the topological

restrictions that the approach imposes. These limitations are based upon the computationally expensive nature of trimming NURBS, its susceptibility to numerical error, and its poor levels of smoothness about the seams of the model. To overcome these shortcomings, the surfaces are often subdivided to allow for the practical application of the NURBS technique in high-end computer graphics (CG) production. DeRose, Kass and Truong (1998) successfully demonstrated their use of NURBS subdivision in their animation *Geri's game* which adopted the approach for modelling the skin on the lead character's head, hands and clothing, which included his jacket, tie, shirt, trousers and shoes. More recently, NURBS are now being applied to complex three-dimensional medical imaging, with researchers such as Zhongke et al. (2003) breaking down curvatures into piecewise rational Bézier curves through the process of voxelisation.

The use of a NURBS surface smoothing algorithm in the test-bed application employed the highly efficient NURBS routines that form part of the OpenGL graphics library. Due to the highly efficient nature of the OpenGL implementation, there was no benefit to developing a bespoke NURBS implementation from scratch. Instead the OpenGL NURBS implementation was tailored to the sheet system by the author to derive maximum visual appeal for the sheet simulation system.

Implementing Curved Surfaces

The most commonly used curve and surface generation techniques are interpolation and data filtering. Three basic methods of designing NURBS surfaces are to:

1. Define a control polygon,
2. Interpolate through a set of points, and
3. Fit a curve passing near a set of points.

Interpolation methods form the most widely used approach, with two variants being available, global curve interpolation and local curve interpolation. Although global interpolation can be susceptible to slight wiggles and oscillations where collinear data points appear, due to the relatively small number of points being used to represent a

sheet, the decision was made that this method provided the greatest suitability to the sheet model in terms of overhead, performance and results.

Following initial system testing, the decision was made to implement a surface fitting algorithm onto the sheet facade. This would enhance the visual appearance of the surface and allow the system to utilise advanced lighting and shading techniques to produce greater realism as the sheet interacts with external stimuli. Of the available choices of surface fitting techniques, the Non Uniform Rational B-Spline curve provided the most effective and stable solution for the sheet application. The system grafts a NURBS layer onto the sheet surface to allow the user the option to enhance the appearance of the sheet as it collides with the polyhedra. This enhanced performance over the use of conventional Bézier curves and allowed the system to use the technique on larger sheet surfaces that would otherwise have been possible.

It was established that a NURBS surface could be effectively mapped on to the mass-spring network by the following assignment of parameters, a knot size of two multiplied by the dimension of the sheet in the x direction (in the parametric u direction) and in the z sheet direction (in the parametric v direction). For example a sheet with dimensions of 5 (in the x direction) by 4 (in the z direction) sheet would have a knot dimension of 10 (parametric u value) and 8 (parametric v value).

The OpenGL graphics library (Neider et al, 1993) was used to implement the NURBS fitting algorithm, and experimental testing conducted by the author was used to evaluate the most effective choice of NURBS parameter (e.g. knots, number of control points) surface smoothing algorithm.

2.3 Sheet Simulation and Deformation

Any of the five animation methods discussed in sections 2.2 can be applied to the modelling of sheets (and a variety of other objects).

Due to the computational expense of generating real-time object deformation, significant compromises in the quality, flexibility and accuracy of the simulation are often required. These compromises include the adoption of simplified environmental dynamics, material properties (such as elasticity, plasticity) and collision detection (Kergosien et al., 1994) to allow the model to run in real-time. Alternatively, "low motion" environments have been created solely for the purpose of generating life-like animation (Volino et al., 1995) but at the expense of real-time interaction.

Montagnat, Delingette and Ayache (2001) produced a positive review of deformable surfaces in their 2001 paper. In their publication they first classified the various forms of surfaces before going on to describe the behaviour of each method and finally detailing similarities with different approaches amongst their peers. However they did acknowledge that deformation models are application dependent (and so cannot be simply classified as one method or another) and that other factors (such as the type of collision detection and overall results based upon the application requirement) can form a more relevant factor than the category of deformation. Cugini et al. (1999) presented an overview of the principal modelling and simulation techniques for non-rigid objects, while also discussing the advantages and disadvantages of each technique. The paper later went on to describe and classify haptic devices for use in synthetic environments before discussing the use of these instruments with non-rigid models.

Deformation research covers a broad spectrum of techniques, such as Mass-Spring networks (Gudukbay et al., 1997; Terasawa, 1996), Finite-Element (FE) based approaches (Pabst et al., 2008, Rodriguez-Navarro and Susin, 2006) and particle systems (Plath, 2000; Desbrun and Gascuel, 1996). Theisel and Kreuseler (1998) have extended the current work on laminates to model 3-dimensional shapes.

Early research into deformation was initially driven by a small number of well-funded research groups that included INRIA (Decaudin et al., 2006), MIRALab (Volino et al., 2005; Cordier et al., 2005), IMAGIS and Caltech, although this position changed about

the turn of the millennium as the cost of computing fell and greater numbers of smaller research groups were able to investigate the subject.

These deformation techniques have been employed primarily to look at the behaviour of objects consisting of material with elastic and plastic properties. The main focus of the resulting animation is the simulation of sheet materials such as cloth fabric (Vassilev et al; 2001); soft substances (James and Pai, 1999; Desbrun and Gascuel, 1995), as well as attempting to produce realistic ‘virtual actors’ for use in animation, teleconferencing and virtual environments (Kalra et al., 1998).

Haiyan and Zhaofeng (2008) describe the use of an explicit Euler method for calculating the Ordinary Differential Equations (ODEs) within the mass-spring based cloth simulation. The motivation for this work was the production of a fast, real-time algorithm for cloth simulation through the use of subdividing the ODE calculations into two segments before merging the results of the computation at the end of the time-step. Mass-Spring systems are used for modelling the cloth utilising Provot’s flexion structure (see section 2.5.1), although their implementation is not sufficiently mature to compete with more established research groups and their respective Provot based implementations, such as those of the University of Geneva’s Miralab Humanoid project team (Volino et al., 2005).

Jiang and Liu (2008) look at common integration methods that are commonly used in the animation of textiles. The authors use accuracy, numerical stability and computational time to draw quantifiable comparisons between the techniques before devising their own universal rules for choosing integration methods based upon differing simulation requirements. In total six methods are assessed which include explicit Euler, Mid-Point, 4th Order Runge-Kutta, Implicit (Backward) Euler, Verlet and IMEX. They go on to discuss the use of mass-spring structures to model cloth and its use of integration methods to calculate the new mass positions based upon forces, acceleration, velocity and time-step. Their experimental work utilises a basic 30 by 30 mesh model to represent a simplified cloth allowing the authors to gain a rudimentary understanding of the benefits and short-comings of the method. The mesh structure is similar to that of Provot’s work with flexion (or interleaving) springs to mimic bending moment within the sheet.

Computer effects in filmmaking are a widely accepted and documented occurrence (Witting, 1999; Barzel, 1992; Foley et al., 1990). Their inclusion into productions is recognised as a great leap in making visually dramatic films at substantially reduced costs. Computer graphics are now employed as a direct replacement to traditional cartoon animation (Barzel, 1997).

Deformation in computer based training has produced promising results. A notable example is found in medical-based simulations (see section 2.6.2) where deformation algorithms play an key role in medical simulations of operations which involve the bending and tearing of skin and organ tissue (Nesme et al., 2010; Zhong et al, 2009; Choi, 2006).

Subsequent sections will now discuss how each of the four fundamental methods identified in section 2.2 are employed in simulations. These sections shall also examine geometric deformation which utilises geometric techniques in order to model sheets. Although this off-shoot of CGI is infrequently applied in commercial applications, it is still worthy of investigation.

2.3.1 Simulation Dynamics - Modelling Ordinary Differential Equations

The standard technique for numerically solving a differential equation is to convert the problem into a finite difference equation. Problems either come in the form of initial value or boundary problems

Mass-spring systems (see equation 2.4) are based upon the force equation:

$$ma = f \quad \text{(Equation 2.4)}$$

that due to its single derivative forms an ordinary differential equation.

Varying levels of accuracy exist for this problem-solving approach, starting with the simple first-order scheme, moving onto the second-order or mid-point method before analysing the advanced fourth-order technique. Each of the three accuracy levels are briefly described below (with some examined in greater depth in Section 3.4, when the

structure of the Sheet Calculation Engines is described). This section introduces the Implicit and Explicit Euler based integration methods that can be used for modelling Ordinary Differential Equation (ODE) problems.

First Order (One-Step) Approaches

First-order, or one-step, approaches offer the simplest and fastest route to resolving differential equations, although the penalty for this speed is accuracy. Time efficiency is gained by the method's modest requirement for information on only one previous step to generate the solution for the next step. For most simple problems the method offers a suitable level of accuracy although in the case of complex mass-spring systems can result in the steady accumulation of numerical inaccuracies. Eventually this leads to a build up of energy that often causes a mass-spring system to oscillate wildly before appearing to explode (Plath, 2000). An example of this instability is demonstrated in figure 3.14. One method of minimising this effect is the increase use of sheet damping to eliminate this build up of energy, often at the expense of over-damping the system, thus producing a flat, or inhibited looking system.

A second and more common solution is to reduce the time-step between calculations, although the cost of this technique is an increase in the levels of processing and slower system performance. There are several one-step methods, which as they increase in complexity, improve the accuracy. The trade-off of this approach is between the increasing work required per step and decreasing the number of steps to span a given range (i.e. through increasing the simulation time-step).

Euler's one-step or constant slope method has limited accuracy and under normal circumstances would not be recommended within a practical application, although it is examined in section 3.4 due to its forming the basis of all subsequent methods.

The Euler-Cromer approach forms an enhancement of the standard Euler method through its differing use of determining the new position of a particle using the velocity of the particle as calculated at the end of the time-step interval, rather than at the beginning.

Second Order (Midpoint) Approaches

The mid-point or second order approach is, as its name suggests, an extension of the first-order technique whereby an intermediate (or mid-) step is used to produce a more accurate prediction of the next time-step. This technique has been known to produce significant improvements in system performance, extending the time-step between frames and enhancing the stability of the results, with minimal increases in the level of calculations required.

The Euler-Richardson approach is an extension of the Euler-Cromer method mentioned in the previous section. The method involves calculating the position of particle during the middle of the interval through determining the mid-time-step velocity of the particle.

Fourth and Fifth Order Approaches

Fourth-order and fifth-order methods provide the highest degree of accuracy for solving differential equations. The most common set of methods form part of the Runge-Kutta family of solutions for solving initial value problems. Within this set, Euler and modified-Euler methods (that include amongst others, the Euler-Richardson technique) form the first order and second order Runge-Kutta methods. The general form of all Runge-Kutta methods for advancing from step i to $i+1$ is presented in equation (2.5) where W_j are constant weighting coefficients and r is the order of the method. The K_j 's represent the estimates of the change in y evaluated at r locations within the range h .

$$y_{i+1} = y_i + \frac{\sum_{j=0}^{r-1} W_j K_j}{\sum_{j=0}^{r-1} W_j} \quad (\text{Equation 2.5})$$

The fourth order Runge-Kutta method has proven itself to be a highly adaptable technique that has achieved widespread use (included within the sheet simulation software) and is covered in greater depth in Section 3.4.1. The fifth order method is an extension to the fourth-order approach that incorporates error-checking functionality into the technique.

2.3.2 Finite Element Analysis and Finite Element Modelling

Finite Element Analysis (FEA) is a numerical method for finding approximate solutions to Partial Differential Equations (PDE) and integral equations. The solution approach is based either on eliminating the differential equation completely (known as steady state problems), or rendering the PDE into an approximating system of ordinary differential equations, which are then numerically integrated using standard techniques such as Euler's method, Runge-Kutta, etc.

When solving partial differential equations, the challenge is to create an equation that approximates the equation to be studied, but yet is numerically stable. This means that errors in the input and intermediate calculations do not accumulate and cause the resulting output to be meaningless. FEA has several ways of retaining stability each with their own advantages and disadvantages.

Finite Element Analysis has proven to be extremely effective at solving partial differential equations over complicated domains (for example, cars, machine components and structures such as cranes); when the domain changes; the desired precision varies over the entire domain; or when the solution lacks smoothness.

For example, in a frontal car crash simulation, it is possible to increase the prediction accuracy in "key" areas such as the front of the car, while reducing it in non-key areas (such as its rear), thus reducing the time and cost of the simulation.

This method breaks down laminar sheets into a series of triangular meshes and/or grid structures with finite mass values at each intersection point.

Finite Element Modelling breaks down an object into a series of small components (known as elements). During the course of a simulation, the Finite Element Analysis will then solve the equations for each element shape within the model.

Each technique varies the scale and range of influence of neighbouring points upon a specific point, with:

1. Energy based methods calculating the total amount of energy present within the cloth through sets of equations to determine the overall shape of the cloth by moving points to achieve a minimum energy state. Energy based equations are used

predominately in static based simulations while force-based methods are employed in dynamic simulations.

2. Force based techniques represent forces among points as differential equations and perform numerical a numerical integration to obtain the point positions at each time-step. Physical techniques generally produce the behaviour of a flexible object that represents cloth, with the degree of accuracy varying from one technique to another.

Bending and deforming a laminar object involves a breadth of engineering-based forces and constraints, modelled using physical laws and principles. Due to the complexity and number of calculations, attempts at modelling object deformation within real-time computer-based environments have resulted in poor calculation performance, limited distortions (Ng and Grimsdale, 1996), restricted scope of object shapes and finite descriptions of performance (Plath, 2000).

Finite Element based deformation tools generally fall into two categories; producing animations for the entertainment industries or being employed to accurately simulate real world events, the differences can be summarised as:

Animation FE toolkits allow interactive manipulation and deformation of objects in a visually acceptable manner. However, these packages employ adapted equations that make an object behave more vibrantly (through the use of enhanced dynamic movement and deformation) than it would in real life so making a scene more visually stimulating than it would otherwise be (Nealen et al., 2006). Examples of applications where FE based laminar modelling would be employed include the modelling of clothes on an animated character.

Finite Element based tools meticulously follow complex sets of mathematical equations to reproduce the object's authentic behaviour at the expense of interactivity with the user, resulting in any simulation being run as batch programs over the course of hours or days. A further drawback is the complexity of the tool which can require a detailed understanding of engineering-based principles by a modeller to allow accurate results to be produced. Examples of FE laminar objects include the modelling of clothes when worn by virtual actors (to determine how the cut of garment will impact its behaviour when worn) and medical applications whereby a layer of skin is sliced open using a virtual scalpel.

2.3.3 Cloth Based Particle Systems

The use of particle systems to animate the behaviour of cloth (Kang and Lee, 2007) was proven to work effectively when Eberhardt et al. (1996) used this approach and managed to reinforce its suitability by developing a system that also took into account cloth specific properties such as hysteresis and measured experimental data.

Plath (2000) provides an excellent account of the issues involved when using mass-spring systems in real-time graphical modelling applied to textiles. Plath describes his method of Cellular Automata where discrete dynamical systems are used and whose behaviour is specified in terms of a local relation, (for example with large classes of continuous dynamical systems defined by partial differential equations). However by his own admission, it is not possible to accurately model the behaviour of a cloth using a set of assigned parameters (for example, for silk), such that an observer could clearly identify the cloth as silk.

In addition, Eberhardt and Weber (1999) detail the effectiveness of different methods for solving the Ordinary Differential Equation (ODE) calculations for their cloth system, with various interpolation methods. Their equations solvers included Runge-Kutta (adaptive fourth and fifth order), Burlisch-Stoer and Lawrence-Livermore methods.

2.3.4 Mass-Spring Sheet Simulations

As discussed in the previous section, mass and spring structures offer a flexible and widely adaptable tool for simulating a broad choice of systems that include cloth, skin, hair, rope, and snakes, as well as deformable and highly elastic/rubber objects.

It was because of this adaptability that the technique was selected as the basis of the sheet simulation system, where its ability to efficiently cater for the effects of internal forces placed upon particles by nearby surrounding particles would instil the system with greater realism when encountering external forces upon individual (or groups of) particles. Forces may include gravity, collision impulses with other objects and/or user manipulation. The adoption of a mesh structure uses each control point as a vertex for a

rectangular mesh. This change over time is due to the influence of external forces and those induced by neighbouring particles.

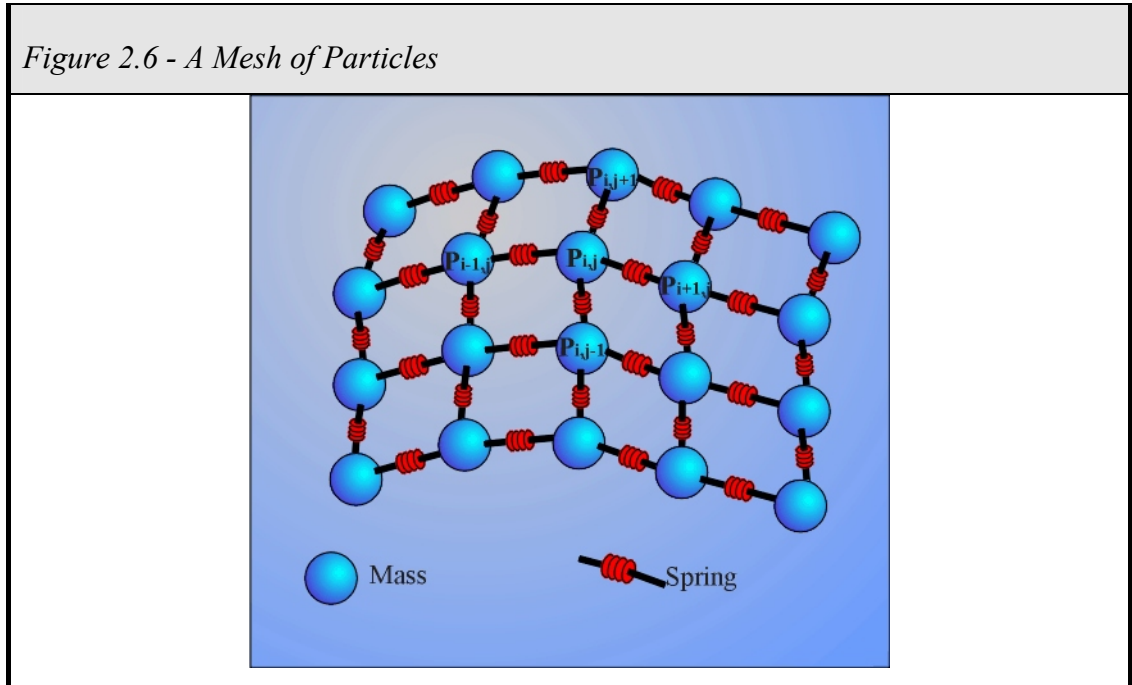


Figure 2.6 illustrates a typical arrangement of a mesh-based sheet particle-spring structure. Any successful simulation will have to consider external and internal sheet forces. If P_{ij} is the location of the mass in row i , column j of the mesh, the force calculation for P_{ij} will have to consider the forces between P_{ij} and the four adjacent locations $P_{i+1,j}$, $P_{i-1,j}$, $P_{i,j+1}$, and $P_{i,j-1}$.

The following section shall now examine the behaviour of these chains of masses and spring as they numerically interact with each other to form a system encompassing multiple degrees of freedom.

Multi-degree of Freedom Deformable Mass-Spring Sheet Systems

One equation of motion exists for each degree of freedom within a multi-degree system. The equations of motion can be obtained via Newton's second law of motion or by using the influence coefficients as described in section 3.6.2. An alternative solution is

to derive the equations of motion of a multi-degree of freedom system by using Lagrange's equations.

For a simple non-damped n-degrees of freedom system (see figure 2.7), the equation of motion can be derived from equations 2.6a or 2.6b given below:

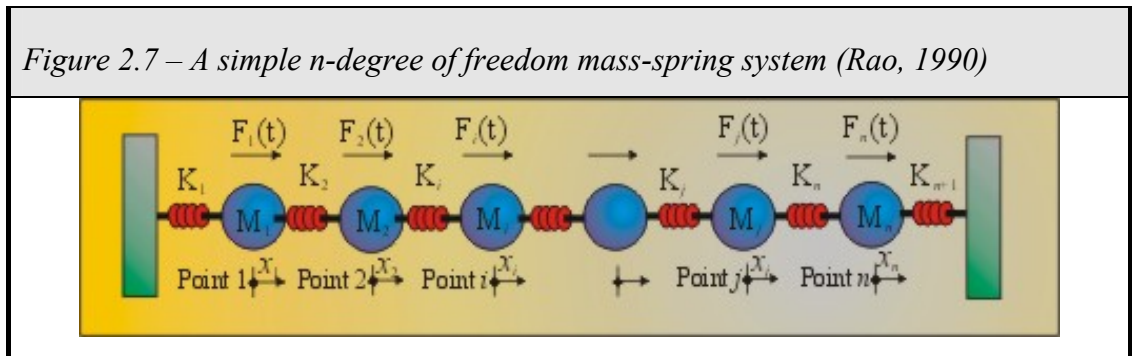
$$m_i \frac{d^2 x_i}{dt^2} = -k_i(x_i - x_{(i-1)}) + k_{i+1}(x_{i+1} - x_i) + F_i \quad (\text{Equation 2.6a})$$

where $i=2, 3, \dots, n-1$

or

$$m_i \ddot{x}_i - k_i x_{i-1} + (k_i + k_{i+1})x_i - k_{i+1}x_{i+1} = F_i \quad (\text{Equation 2.6b})$$

where $i=2, 3, \dots, n-1$

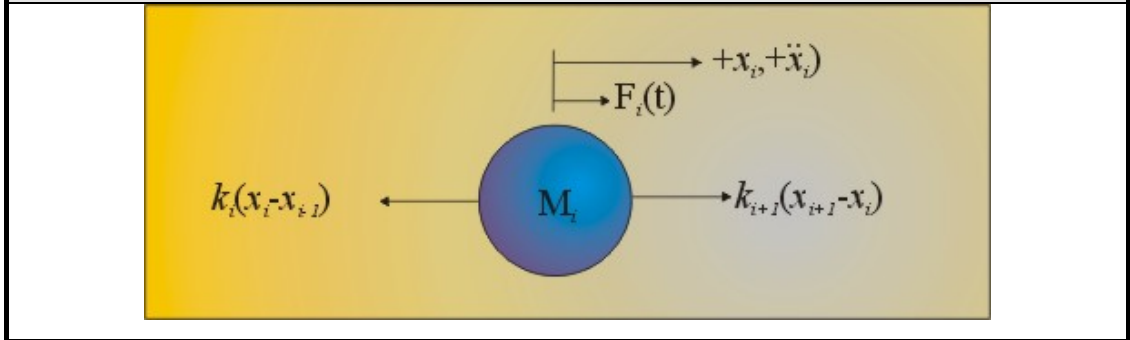


The equations of motion for masses m_1 and m_n (see figure 2.8) are derived from equation (2.6). In the case of m_1 this is achieved by setting $i = 1$ and $x_0 = 0$ resulting in equation (2.7), while with m_n , this is achieved by setting $i = n$ and $x_{n+1} = 0$ to produce equation (2.8).

$$m_1 \ddot{x}_1 + (k_1 + k_2)x_1 - k_2x_2 = F_1 \quad (\text{Equation 2.7})$$

$$m_n \ddot{x}_n - k_n x_{n-1} + (k_n + k_{(n+1)})x_n = F_n \quad (\text{Equation 2.8})$$

Figure 2.8 – Forces placed upon Mass M_i



Mass-Spring Integration Methods

Two approaches exist for performing the integration of mass-spring systems, with each approach encompassing its own advantages and disadvantages for their use within simulations, depending upon the characteristics of the object being modelled, realism, available computational power and level of real-time interaction. These characteristics are examined in this section.

a) Implicit Euler Integration

Implicit Euler integration works by replacing the forces at time t by the forces at time $t + dt$ as demonstrated in equation (2.9a) and equation (2.9b).

$$v_i^{n+1} = v_i^n + F_i^{n+1} \frac{dt}{m} \quad (\text{Equation 2.9a})$$

$$x_i^{n+1} = x_i^n + v_i^{n+1} dt \quad (\text{Equation 2.9b})$$

The advantage of this substitution is that it enforces stability by ensuring that the new positions of the masses are not blindly reached as they correspond to a state whereby the force field is coherent with the displacement found. This is achieved through the use of an approximation of the force at the next time-step that introduces a level of feedback into the integration process. Through assuming that the forces within the structure are kept constant (regardless of the time-step size) it avoids any instabilities that may arise. Compared to explicit schemes which move from step to step only

knowing the initial conditions, implicit schemes attempt to land on the next position correctly.

Desbrun et. al. (1999) have demonstrated the use of an implicit integration engine combined with a predictor-corrector approach and inverse dynamics process to model of a silk scarf in a virtual environment. The stability inherent within implicit integration is owed to two principal factors, these being the addition of artificial viscosity and the filtering of the force field. Implicit integration is known to “implicitly” add a set of forces to the internal forces of the system that are proportional to both stiffness and time-step size that act as a damping force producing a level of artificial viscosity, ensuring that each mass will inherit a tendency to follow the local displacement of its neighbours, hence avoiding wild local instabilities. In animation, a simulation with a fixed time-step dt may only handle deformation frequencies up to a specified limit. Once reached, the sampling rate will produce instabilities into a system. By automatic filtering of the force field to remove high frequencies from the system, the simulation will remain unconditionally stable.

For all its inherent benefits to stability and damping, the implicit approach has several drawbacks that limit its use in mass-spring systems.

The complexity of having to solve a linear system results in either a significant computational overhead as any linear system will have non-zero spring lengths, the forces will not remain linear (preventing real-time use as the system forces will need to be recalculated for each-time-step). To overcome this, the adoption of a significant assumption is needed, which requires a temporary spring length of zero for each sheet spring. Then, to compensate for the errors that are produced during the integration phase, a correction factor would be needed to correctly calculate sheet momentum. Although linear momentum is maintained through this technique, angular momentum is not. This short-fall is then compounded as the stiffer the spring, the greater the loss of angular momentum. In an attempt to compensate for this loss, each mass would need to be given a “push” (usually through use of a linear approximation) to induce angular rotation. This technique, initially pioneered by Baraff and Witkin (1998), has been used successfully by numerous researchers for modelling cloth (Desbrun et al., 1999; Meyer et al., 2000). The loss of rotational stiffness, however, makes it difficult to model

laminate materials other than those with insignificant levels of bending stiffness, such as cloth.

b) Explicit Euler-Based Integration

In the majority of simulations, explicit integration using Euler integration is found to be a more effective technique for a single time-step, due to its simplicity and flexibility in reproducing dynamic behaviour.

Due to the shortcoming of implicit integration (in relation to the loss of rotational stiffness) which limits its implementation in real-time modelling, other than for modelling materials with low bending moment, explicit integration forms the best, broad-based algorithm for real-time simulation. Any shortcomings of explicit integration can be overcome through the use of time-step manipulation, use of different levels of integration and novel sheet-structures.

The use of implicit and explicit algorithms is examined in further depth by Wu in the International Journal of Clothing Science and Technology (Wu et al., 2003).

The use of explicit integration within the (MaSSE) system resulting from this research is examined in greater depth in the following chapter.

Verlet Integration

The Verlet method of integration does not require velocity to be stored as it does in the similar Euler integration method. The velocity component is used to integrate all field forces such as spring-force and gravity.

The advantage of the method is that, because it does not explicitly store the velocity of each mass particle, it is much more stable than the Euler method when particles are manipulated to satisfy constraints within a simulation. Therefore this allows for greater stability during resting and collision contacts within a simulation.

However, the trade-off of this stability is accuracy, and although it has proven to be an effective technique in computer games where stability and low computational overhead

are key, the decision was made by the author of the work to retain the Euler method for integration due to its greater flexibility and adaptability in producing life-like results.

Harada et al (2007) successfully demonstrated the use of the Verlet integration in their combined cloth and fluid simulation model. However by focusing their research upon combining cloth simulation with fluid dynamics, they constrained the movement of the cloth dynamics which better catered for the use of Verlet integration.

2.3.5 Geometric Sheet Modelling

Geometric modelling of static objects does not consider the physical properties of cloth but instead focuses upon appearance, particularly in complex areas such as folds and creases where they are reproduced using geometrical equations. Geometric techniques do require significant levels of user participation and intervention and consequently are sometimes regarded as little more than an advanced drawing tool.

Deformable surface geometry is administered by two elements, the representation of the model's plane and the rules that are used to govern its ability to change shape.

The surface geometric representation is controlled by two key criteria, shape description and deformation description. The former outlines restrictions that may prevent an object from forming anything other than simple shapes (such as ellipsoids), shapes of restricted topology (i.e. spherical Fourier descriptors) or shapes with independent topology such as mass-spring systems. Deformation description works by deforming the embedded space of a model, through, for example the application of a global transformation such as an affine transformation.

Deformable surfaces are divided between continuous and discrete models. For discrete models the geometry is either only known at a finite set of points or for continuous models where (for computational reasons) the system is discretised but also offers the ability to compute differential quantities, such as surface normal or curvature along almost any point of the plane.

When an object has been broken down into smaller manageable components, then constraints must be placed upon the complex model to limit its movements and make it

move in a physically plausible manner. It is these constraints that prevent the modelled system from simply acting like a disconnected group of small objects and more like the single large object that was originally intended. An illustration of this technique is when attempting to model a rope as it is dragged along the ground at one end in a wave-like pattern. Breaking the rope into smaller components allows the system to be modelled in a simpler form, however if the rope is not constrained then it will appear to behave more like a chain arrangement rather than a homogenous structure.

The use of geometry for modelling sheet deformation was employed predominantly prior to the widespread use of Finite Element and numerical methods such as mass-spring systems. The reason for its early growth was due to its efficient method of producing smooth surfaces with significant levels of computational efficiency. However these models were limited in their behaviour due to their failure to take into account the internal forces that were being applied within the sheet model models. As computational capabilities grew, reducing the negative impact of employing finite element and numerical based methods, geometric systems have fallen out of favour and are now used predominantly in restricted animation applications where a pre-defined movement, combined with high levels of visual accuracy are required.

2.4 The Relationship between Animation & Simulation

Animation is described as the study of movement (Vince, 2000) – moving characters, moving objects, moving cameras, moving lights and moving special effects. Whilst traditional animators rely upon their drawing skills to empower a character with life-like movement, computer animators rely upon software to realise movement.

Animation is described as the process of:

“Generating repeated renderings of a scene, with smoothly changing viewpoint and/or object positions, quickly enough that the illusion of motion is achieved.”

(Neider et al., 1993)

Foley et al. (1990) describes the process of animating as quite literally, ‘to bring to life’, going further to describe how animation is not just synonymous with motion, but covers all changes that have a visual effect. These include:

1. Time-varying position,
2. Shape, colour, transparency, structure, and texture of an object, and
3. Changes in lighting, camera position, orientation, focus and changes in rendering technique.

Physical Simulation is about simulating physical behaviours within the virtual world of computer animation using various physics based laws.

When a simulation based system has been broken down into manageable elements and any necessary real world constraints applied, it is ready to evaluate against not only external factors such as foreign bodies or forces as they collide and influence the model, but also internal forces from within the object itself. This is because the behaviour of one area of the entity may influence the performance of the whole structure, for example if one area of the sheet falls off the edge of a face under the effect of gravity it will attempt to drag the rest of the sheet with it.

Typical forces in the simulation of physical phenomena include:

1. Acceleration due to gravity i.e. a constant downward force that is directly proportional to the object-mass and which acts upon the centre of mass.
2. Damping forces which are opposite and proportional to the body's velocity which act to resist its motion. In reality, damping forces remove energy from a system and dissipate it as heat. A viscous damping force is proportional to velocity while a quadratic force is proportional to the square of speed. Ignoring the effects of air disturbance, air resistance is approximately quadratic.
3. Elastic springs that are used to connect two bodies with a force proportional to the displacement of the spring from its rest length (i.e. Hooke's law). This is covered in greater depth in chapter 3.
4. Geometric constraints of real bodies. These are used as a constraining force upon a simulated body to restrict the level of movement (or degree of freedom) allowable by the object. An example is a pendulum as it swings around a fixed point its movement is restricted by the pendulum arm.

Although there is a need and significant scope for enhancing the performance of dynamic models, Ronan Barzel (1997) in his interesting paper on faking dynamics of ropes and springs described how “real” (or exact) dynamics are rarely used in their accurate form during the animation process as they infrequently provide the desired visual stimulation for the entertainment industry. Instead it is common to modify the behaviour of the objects or characters within animation to match the storyline of the animation.

Barzel illustrated the differences between animating a scene with a fictional character, utilising ‘exaggerated’ dynamics and the accurate simulation of characters and objects using ‘real’ dynamics. The justification for using ‘fake’ as opposed to ‘real’ dynamics is to help convey story, mood and character, thus helping to direct and hold the audience's attention. Due to these factors, the dynamics involved within the animation would vary from object to object, scene to scene, pose to pose and frame to frame in order to meet the needs of the moment. Breen in his 1997 paper describes additional examples of modifying dynamics for aesthetic appeal.

A further hindrance to utilising ‘real’ dynamics in an aesthetic (animated) simulation is that the control methods used in dynamics do not readily lend themselves directly to non-physical motion, and quick-per-pose and per-frame editing. This is an important

feature required by animators, which rapidly forces the interaction speed between the system and the animator to become an issue. Based upon this, it is clear that the requirements for animation-based deformation and dynamic movements vary significantly from techniques employed in producing realistic or 'life-like' virtual environment based simulations.

2.5 Applications and Implementations

This section shall discuss the needs and requirements required by applications when either directly simulating the behaviour of laminar objects such as cloth, as well as indirect objects that either apply the same methods as those used in laminar modelling (e.g. human skin modelling) or that interface with these simulations such as virtual humans.

2.5.1 Cloth Simulation and Deformation

The mechanical behaviour of a fabric is not simply a dynamics problem, it is also linked directly to the nature and molecular structure of the fibre material that constitutes the cloth; especially the way that these fibres are arranged in the fabric structure because these will determine the stiffness in different directions. Fabric fibres on a cloth surface may be arranged in several ways, such as those of woven fabrics, knitted fabrics and non-woven fabrics. For example, woven fabrics contain threads that are orthogonally aligned and interlaced alternately using different patterns (such as plain or twirl). The threads in knitted fabrics are curled along in a given pattern with the twirls interlaced on successive rows. Non-woven fabrics contain no threads but instead the fibres are arranged in an unstructured form, such as in paper fibres.

Due to their stiffness, relative thinness and easy production, woven fabrics represent the most common form of garment production. Knitted fabrics with their high degrees of elasticity and looseness are often employed in woollen garments. The fabric structure greatly influences the mechanical behaviour of the material, which is determined (to a large degree), by the following properties:

1. The nature of the fibre (i.e. wool, cotton, synthetic etc.)
2. The thread structure (i.e. diameter, internal fibre and yarn structure)
3. Thread arrangement (i.e. woven or knitted, and the pattern variation)
4. Pattern properties (i.e. tight or loose).

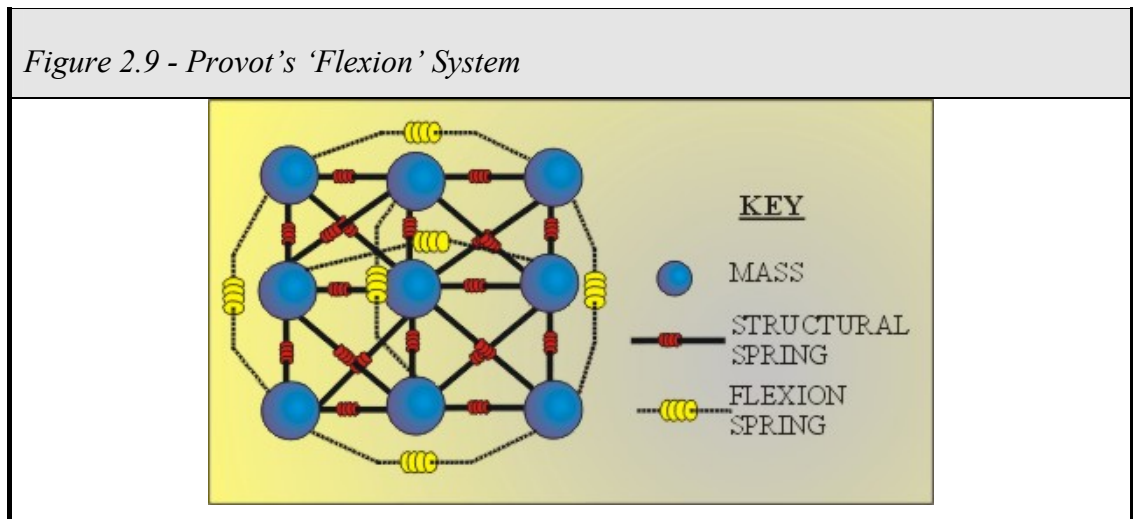
These parameters combine to form the overall stiffness, bending ability and visual appearance of a material.

One variation from the traditional theory of elastic behaviour that cloth introduces is the assumption that the orientation of the material has no effect upon its mechanical properties, otherwise known as isotropy. This, however, is not the case with cloth as any deformation will be heavily influenced by its relative orientation to the fabric thread.

According to Maurizio Vecchione during the panel discussion at Siggraph '97 entitled "Current Challenges in Cloth Design, Modeling and Animation", over 70% of apparel is at least partially designed on computer (Breen et al., 1997). This creates a great opportunity for researchers not only to gain funding by commercial and government bodies to investigate this area, but also in the process to significantly increase the probability of producing a commercially viable product. One of the most exciting applications of virtual fashion is the ability of designers to draw a basic concept of a garment, give it material properties and immediately watch the behaviour of the clothing as it is worn by a virtual model on a virtual cat-walk without even leaving his/her computer and all within a fraction of the time, cost and effort required using traditional methods. This line of research has been strengthened through the ability to scan the body dimensions of real actors thus providing realistic models upon which the clothes are worn. Volino et al. (2005) examined the development of garment simulation in their 2005 review paper with research conducted by Fontana, Rizzi and Cugini focusing upon employing CAD technologies to produce industrial applications (Fontana et al., 2006; Fontana et al., 2005). The work of Liu et al. (2009), Sayem et al. (2010) and Fontana have clearly demonstrated the suitability of CAD systems for producing robust applications for the garment industry. Draping models have formed an important element of cloth modelling with the drape characteristics forming an integral component of any successful system. To achieve this aim the Faculty of Textile Science and Technology at Shinshu University, Japan has successfully demonstrated the effectiveness of their system for skirt modelling (Dai et al., 1998), using the Kawabata Evaluation System (KES) that is used to measure the mechanical properties of cloth (Volino et al., 2009).

Most cloth simulations are comprised of two-dimensional structures that employ varying approaches to reproduce the complex, resource-intensive, rotational stiffness calculations that are required during sheet bending (Breen et al., 1997; Gudukbay et al., 1997; Ng, Grimsdale, & Allen, 1995).

Provot (1997, 1995) best illustrates in figure 2.9 an effective approach that uses two-dimensional sheet structures. Here, to allow for moment forces in his model, special “flexion” springs are appended to this “conventional” sheet structure. These springs behave in a different way to the conventional structure-based springs that cater specifically for compression, tension, and shear-based calculations.

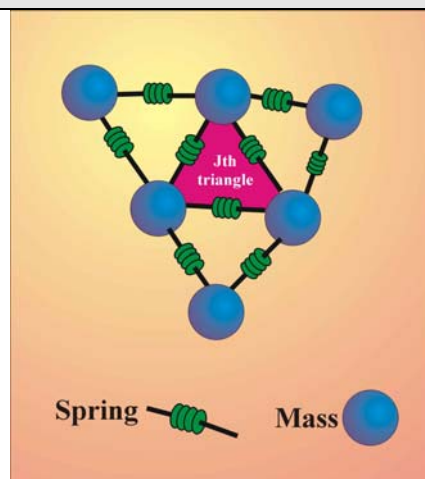


The work of Provot is examined in greater depth in Section 3.3, drawing comparisons between his method of cloth simulation and those presented in the present work. Provot's work has been extended by numerous researchers, with Zhang and Yuen (2001) supplying more complex variations of the flexion system through their use of tension and crossed spring structures. Confusingly they refer to their system as a multilevel mesh system when in reality it is a two-dimensional mesh system with overlapping springs.

An equally effective technique was proposed by Ng, Grimsdale and Allen (1995) who developed an energy-based method of simulating draped and suspended cloth using a point-based environment. Their approach utilised modest PC hardware to generate their results and it is this lack of suitable hardware that may be partially attributable to the non real-time nature of the system. A variable grid-size system (referred to as the Multigrid Method) is used to adapt the coarseness of the sheet thus maximising the visual dexterity of the model in active areas of the sheet whilst minimising the performance overhead.

Cloth simulation initially attempted to simulate the draping behaviour of regular shaped blankets and table clothes as they lay and in later cases dropped onto surfaces. As the simulations became more complex, there was a move towards simulating clothes worn by dummies and in the most advanced cases, by walking avatars. The Humanoid projects, under the leadership of Thalmann and Magnenat-Thalmann (Volino et al., 2005) provided the first (and most revolutionary) examples of this work. This move from simple simulations to fully fledged simulations of garments highlighted some of the additional complexities that required consideration. These included the reality that clothes are formed from complex shapes which are often made from many separate components that must take into account the fabric weave, the stitching that adjoins these pieces and the angle that a piece interconnects to its neighbouring pieces. In addition, the shape of the material is often determined by the number and shape of collisions that occur not only between the wearer and the clothing, but within the cloth itself. The Thalmann's work in the Humanoid projects uses mass-spring systems that form irregular triangles (see figure 2.10) employing Newton's motion equations. These are resolved using second-order and mid-point methods for solving ordinary differential equations.

Figure 2.10 – The Thalmanns Humanoid Cloth System



Ng and Grimsdale (1996) presented a wide ranging survey paper detailing nineteen cloth-based modelling techniques based upon their principal theoretical method. This included examinations of geometrical, hybrid and physical-based techniques that had

been predominantly developed over the previous six years. The review examined the techniques employed by research groups to mimic the cloth, as well as outlining how the cloth systems were demonstrated using the examples created by the investigators themselves.

Eberhardt et al. (1996) use particle systems to model the behaviour of animated cloth, also taking into account cloth specific properties such as hysteresis and measured experimental data. Eberhardt and Weber in their 1999 paper detail the effectiveness of different methods for solving the Ordinary Differential Equation (ODE) calculations for their cloth system, with various interpolation methods. Their equations solvers included Runge-Kutta (adaptive fourth and fifth order), Burlisch-Stoer and Lawrence-Livermore methods. Plath (2000) on using interacting particle systems to model textiles provides an excellent account of the issues involved when using mass-spring systems in real-time graphical modelling.

A further industry-led application of cloth-modelling was presented by Aono et al. (1996) who presented a system for accurately simulating the behaviour of *woven cloth composites* which are used to reinforce aircraft structural elements. These resin-impregnated woven sheets are in reality darted (i.e. have areas of the cloth removed or stitched to avoid gaps and wrinkles) but which result in anomalies when modelled within a CAD system. Their proposed algorithm was devised specifically for overcoming these incongruities in the final three-dimensional ply.

Baraff and Witkin (1998) presented a benchmark paper at SIGGRAPH 98 where they investigated the relationship between the size of simulation time-steps in cloth animation and numerical instability. The paper, entitled 'Large Steps in Cloth Simulation' describes the use of implicit differential equations to produce cloth draping of a 2600 node cloth at a running time of 2 seconds per frame utilising an SGI Octane processor. Although Baraff and Witkin demonstrated positive results for the use of implicit integration with cloth, they were nevertheless still required to produce an adaptive step-size facility in order to retain system stability. The relation between simulation time-step and numerical stability is examined in Chapter 3.

Researchers at Caltech in the USA and the French iMAGIS group (Meyer et al., 2000) extended the work of Baraff and Witkin (1998) producing impressive results of cloth modelling as worn by virtual actors. This approach is also based upon the use of

implicit integration techniques to extend the time steps required for cloth modelling. It also included removing the shortcoming of solving a large linear system during each time step. The final solution utilises an amalgamation of implicit and explicit integration combined with post-step correction to produce efficient cloth simulation in real-time.

Hutchinson (Hutchinson et al., 1997) has produced an effective technique for employing mass-spring networks to simulate the behaviour of sheets, cloth and string using an adaptive level of refinement across the model. This allows the model to produce the most effective and realistic representation of the simulation whilst ensuring that the model is not wasting computational resources over-modelling dormant areas of the sheet. Although utilising simple collision detection methods with limited environmental parameters (for example, lack of friction), Hutchinson's approach has shown promising results. However, the technique does involve an additional computational overhead when compared with traditional uniform approaches. This disadvantage was overcome through the combination of the limited use of environmental and simulation dynamics, a non-real-time implementation and the use of high specification computational hardware.

Ling (Ling, 1997; Ling et al., 1996) has investigated the behaviour of cloth as it deforms whilst encountering aerodynamic forces. Initial demonstrations involved animating a curtain being blown by the wind, with later examples simulating a skirt when encountering air-flow.

Mass-Spring systems are used to model laminar materials under a variety of conditions. These include changes to their configuration over time due to external influences such as gravity (Eberhardt & Weber, 1999; Howlett & Hewitt, 1998), manipulation by virtual actors (Baraff & Witkin, 1998; Provot, 1997) or wind (Dai et al., 1998).

2.5.2 The Kawabata System's Fabric Material Properties

The complexity of fabric modelling is illustrated by the Kawabata Evaluation System (Volino et al., 2009) which is used to aid the experimental measurement of the elastic properties of fabric material. Five experiments are used to produce fifteen curves that in turn determine twenty-one parameters that depict the behavioural properties of the cloth material.

The experiments are briefly described below (Volino and Magnenat-Thalmann, 2000):

1. The Tensile Test - The extension of a standard sized (5cm by 20cm rectangular) cloth piece at a constant speed of 0.1 cm/s or 0.2 cm/s extending in opposite corners with the remaining two corners fixed. This is used to calculate the rest elongation, maximal elongation, elongation ratio, traction energy, recovery energy, resilience, and linearity.
2. The Shearing Test - Similar to test (a), a standard cloth piece is moved transversally with the shearing stiffness (taken at a 2.5° derivation) and two hysteresis amounts (taken at 0.5° and 5.0° derivation) being computed.
3. The Bending Test - A thin fabric rectangle (1 cm wide by 20 cm long) fixed along its length is rotated along its direction and the distance between the two lines of attachment are adjusted to preserve the curvature as being a cylinder section. This test yields the flexion stiffness and the hysteresis at 1cm curvature radius.
4. The Compression Test - This compresses a cloth sample between a flat surface and a 2cm² cylinder to determine the rest thickness for 0.05 KPa pressure, the compressed thickness for 5.0 KPa pressure, compression energy during compression, recovery energy, the compressibility ratio, the resilience, and the linearity.
5. The Friction Test - Measures the roughness and the friction on a fabric surface at constant tension, using one and ten rods of 0.5 mm section and 5 mm length respectively, applied with a force of 0.5 N, to calculate the roughness, the friction coefficient and its variation between the two cycles.

Based upon these tests the mechanical properties of a cloth, such as, for example, plain wool-polyester of 125 g/m² density can result in the approximate values as stated in

table 2.1. The following values also act as a good example of typical values that form common fabric types (Wu et al., 2003). The symbols relate to the variables in equations 1.3 and 1.4.

<i>Table 2.1 – Example Mechanical Properties of Wool-Polyester Cloth</i>	
Parameter	Value
Young Modulus, Warp (D_1)	3600 N/m
Young Modulus, Weft (D_2)	3200 N/m
Young Modulus, Diagonal (D_s)	1200 N/m
Rigidity Modulus	35 N/m
Poisson Coefficient, Warp (V_1)	0.18
Poisson Coefficient, Weft (V_2)	0.16
Flexion Modulus, Warp (H_1)	5.1 μ N/m
Flexion Modulus, Weft (H_2)	4.4 μ N/m
Flexion Modulus, Diagonal (H_s)	3.7 μ N/m
Flexion Rigidity	1.5 μ N/m

It is common to see a high variation in the above values for different fabric materials, particularly for knitted fabrics, however table 2.2 details the ranges have been observed for most common fabric types (Volino and Magnenat-Thalmann, 2000).

<i>Table 2.2 – Common Value Ranges for Fabric Materials</i>	
Parameter	Value Range
Young Modulus, Warp (D_1)	2000 - 6000 N/m
Young Modulus, Weft (D_2)	1000 - 5000 N/m
Rigidity Modulus	20 - 60 N/m
Flexion Modulus, Warp (H_1)	5 - 15 $\mu\text{N/m}$
Flexion Modulus, Weft (H_2)	4 – 10 $\mu\text{N/m}$

The Kawabata experiment demonstrates that even though it is possible to categorise the properties of laminar materials with values to demonstrate their real-world strength and dynamic behaviour (Volino et al., 2009), due to the limitations of all numerical methods in simulating complex real-world objects, it is not possible to produce a direct correlation between these real-world parameter values and those values represented in a simulation. A complex calibration exercise is often required to conduct a deformable sheet simulation with material properties only covering a some of the influential variables, with simulation specific parameters such as time-step size, integration method and environment (and sheet) damping having an equivalent level of influence on the simulation outcome.

A simpler alternative to the Kawabata Evaluation System (KES) that also measures the mechanical behaviour of cloth using standardized protocols is the FAST method that is also based upon the experimental measurement of strain-stress curves for elongation, shearing and bending on normalized samples of fabric. These then allow different representations of the cloth surface mechanics to generate a virtual reproduction of the cloth behaviour.

2.5.3 Skin Deformation and Virtual Actors

The portrayal of autonomous or ‘virtual’ actors for modelling the characteristics of creatures that form different shapes and sizes is starting to play an ever more important role in the formation of both virtual environments and real-time animations. One aspect of this research directly applicable to the current work is the simulation of skin as it stretches and bunches on actors as they move their limbs. This deformation process is centred on the internal bone structure of the actor, also muscle formation, definition of fat and the shape of the animal. The Humanoid project operated by Thalmann and Magnenat-Thalmann (Boulic et al., 1995) covered the subject of facial, hand and skin modelling in extensive detail, as examined in section 2.10.

Wilhelms (1997; Wilhelms and Gelder, 1997) demonstrated a simple and highly generic algorithm based upon a marching cubes approach to model the physical behaviour of animals and humans, including specific limbs such as human hands and legs. Through use of a jointed endo-skeleton (i.e. “inside skeleton”) which is moved by muscles and covered with a flexible skin, realistic and life-like animal movement is possible. This work, however has tended to concentrate solely upon aesthetics as opposed to autonomous actor movement. Scheepers et al. (1997) also attempted to reproduce highly accurate models of human shape and movement, adopting an artist’s approach through the gradual build-up of actors from the skeletal frame, adding musculature before the application of skin. Skeletal movement is controlled through the use of articulation variables (or avars) while like Wilhelms, adopting the use of ellipsoids to represent muscle bellies.

Free Form Deformations (FFD) together with shape interpolation form two of the most common approaches to simulation of the skin deformation process as actors move their limbs and change facial expressions. Shape interpolation is particularly favoured for use in facial animation while body modelling and deformation is usually achieved through the use of skeleton-driven deformation techniques. A notable exception to this is the work of Nedel and Thalmann (1998) who utilise spring mesh dynamics to simulate the surface deformation of muscles, with a modelled skin cross-section then being reshaped through use of ray-casting to determine the maximum displacement of the underlying tissue. Lewis, Cordner and Nickson (2000) have put forward a generic

approach known as Pose Space Deformation that caters for both facial and body deformation.

Turner and Gobbetti in their 1998 paper discuss the innovative Layered Elastic Model Animation System (LEMAN) for modelling highly flexible, deformable characters. Their approach uses layered models that are comprised of four levels; the skeleton, the stick-figure representation of the actor's links and joints; bones which define the geometric shapes of the bones upon the skeleton; fat layer comprising of springs that connect the muscle (attached to the skeleton) to the skin layer; skin layer that forms a continuous elastic surface.

2.6 Graphics Processing Units

As computer games become more advanced and user expectations continue to grow, computer designers have turned to the Graphics Processing Unit (GPU) to relieve the stresses and bottlenecks that are being placed upon the computer's Central Processing Unit (CPU) and motherboard.

GPU based programming works by moving the complex three-dimensional calculations used in games and simulations from the CPU to the GPU thus reducing the loads on the CPU (and the system memory), alleviating the amount of data traffic that is passed between the CPU/main memory to the graphics card, utilising the advanced parallel processing capabilities present within a GPU and making better use of the available resources on the graphics card, in terms of the GPU and graphics card memory.

According to Song and Grogono (2009), GPU programming has a number of benefits that include:

1. It releases the computational power of the relatively inexpensive GPU available to act as a co-processor for simulations.
2. The recent and wide adoption of GPU programming has led the availability of a broad number of algorithms for use across a wide range of applications.
3. Parallelisation of program code can substantially boosted system performance.
4. The development of new high-level languages, libraries and toolkits has broadened the scope of GPU programming.

However the adoption of GPU based development does come with a number of shortcomings that include:

1. GPU design is centred on enhancing the game-play of computer games, rather than improving computational simulations.
2. The assembly language of Graphics Cards can be difficult to use.
3. The GPU programming model is unusual.
4. GPU programming paradigms are specifically tied to computer graphics.
5. The programming environment is tightly constrained which limits its flexibility and use.

6. Due to its rapidly evolving nature, its feature set has expanded significantly.
7. Large aspects of GPU design are kept secret by manufactures.
8. All existing CPU produced software will have to be rewritten specifically to run on a GPU with no simple 'porting' process available.
9. The GPU architecture is poorly suited to traditional, sequential "pointer based" programming code.

Therefore it is easy to surmise that GPU programming is relevant to computationally expensive systems such as games, where even a minor improvement in performance can have substantial benefits (in terms of game-play, sales etc.). GPU programming offers exciting possibilities for its adoption in computer based research however there are a number of significant short-comings, least of all being that all existing software will not only have to be re-programmed from scratch but that the structure of the system will be compromised to cater for the foible of GPU development.

2.7 Discussion

The motivation of this work is the desire to efficiently simulate stiff laminar materials in a real-time capacity employing the use of novel methods. This chapter has reviewed a number of possible computational methodologies for achieving this goal.

Their strength and limitations (in relation to the research objectives) of the four principal methods of modelling laminar materials are summarised in table 2.3.

<i>Table 2.3 – Suitability of Deformation Methods to Hypothesis Objectives</i>				
	FEA	Mass-Spring	Particle	Free Form Deformation
Real-Time	Limited	✓	✓	Limited
Ability to Represent Bending Moment	✓	✓	✗	✗
Computational Efficiency	✗	✓	✓	✗
Environmental Effects (Heat and/or thermal softening)?	✓	✓	✓	✓
Flexibility (ability to modelling different simulations and applications)	✓	✓	Limited	✗
Degree of Accuracy	High	Aesthetic	Aesthetic	Aesthetic

From the analysis summarised in table 2.3, it is clear that the mass-spring approach appears to represent the most promising technique of reproducing the effects of Bending Moment force in a real-time environment.

If mass-spring systems are to be used to model stiff laminae in real-time then an efficient method for simulating the effects of bending moments must be found.

This work shall assess the effectiveness of three-dimensional mass-spring systems arranged in a two-layer lattice framework for reproducing the visual behaviour of laminar materials in a computational efficient real-time environment.

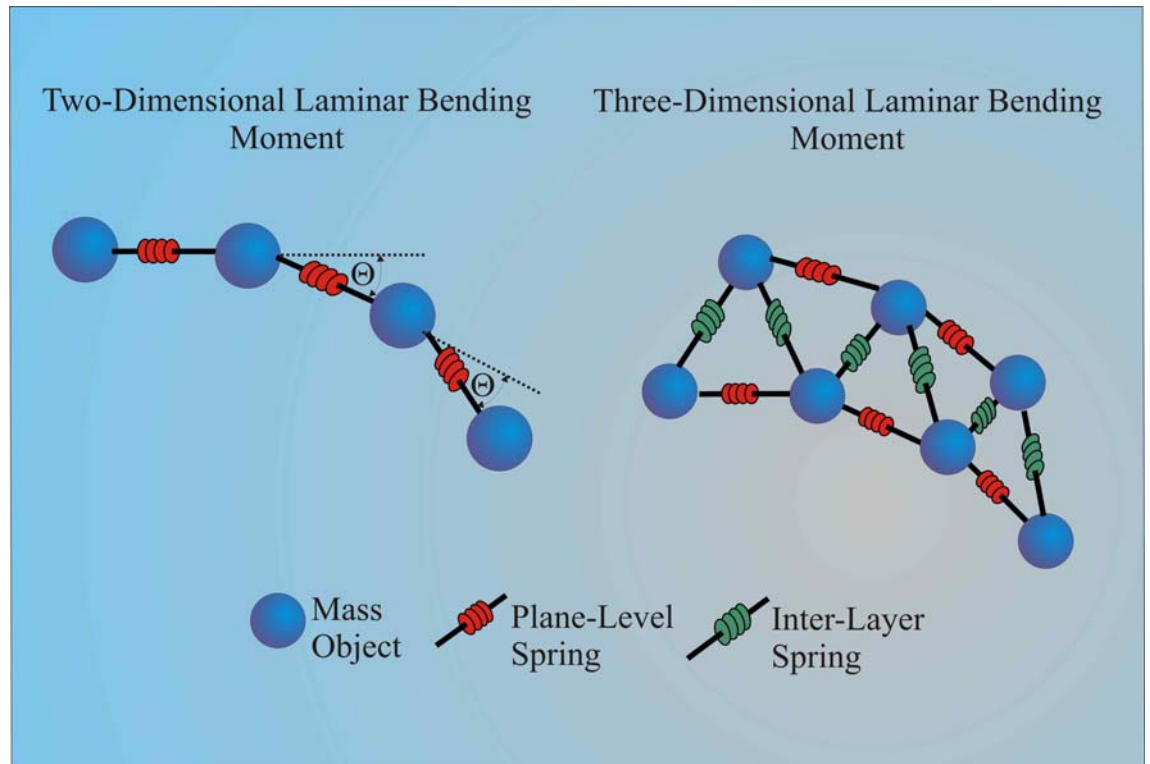
To achieve this aim, a Mass-Spring Simulation Engine (known as MaSSE) has been developed that will form part of an application that is referred to as 'Aurora'. The Aurora application will also embody a 'World on Window' (WoW) front-end interface that enables a user to visualise a simulation and to manipulate its parameters.

To automate the production of the complex sheet structures, a second application has been produced that is referred to as 'Lightning'.

Traditional mass-spring research tends to employ 2-D structures (such as those illustrated in figure 2.11) with complex rotational stiffness calculations to simulate bending moment (Gudukbay et al., 1997).

This is achieved by implanting each mass with a horizontal axis component that identifies a horizontal plane for the mass. Then, through a complex set of calculations (that are performed in every simulation time-step), a comparison is made between the angles of the two opposite-end springs that are connected to each mass. If their angle value (indicated by θ in figure 2.11) is greater than an arbitrary preset value, restrictions are placed on the angle of the second spring to ensure that θ remains at the preset rotational value.

Figure 2.11 - Rotational stiffness and MaSSE bending model.



As a breakaway from the employment of complex, resource-intensive, rotational stiffness calculations during sheet bending (Breen et al., 1997; Gudukbay et al., 1997; Ng et al., 1995; Provot, 1997), this work will employ three-dimensional mass-spring structures to enable computation of the bending moment force. Together with experimentation in our various sheet structures, these lead to an efficient alternative to current approaches.

Furthermore, the use of a third dimension (depth) allows an overall greater level of customisation, because the interlayer cross-springs can then be modified to cater for different levels of modular stiffness.

Figure 2.11 illustrates how the MaSSE approach uses a simpler piece of code to calculate sheet behaviour that is executed many times, rather than use a complex (rotational stiffness) algorithm that is involved fewer times per frame. This way,

through careful optimization of the system, significant improvements in sheet simulation will be achieved.

This system could have several advantages over conventional 2-D models such as those proposed by Provot (1996). These include the fact that:

1. Any arrangement of sheet (or full three-dimensional) models may be developed and tested using the technique.
2. The structure follows a closer representation of real world problems in which any bending or deforming modelling represents ideas based upon three-dimensional laws and mathematical principles.

To enhance the flexibility of the system, two classes of springs have been devised to differentiate between shear and bending-moment force that are characterised by plane-level and interlayer springs.

The described mass-spring approach differs from conventional finite-element-based systems (Pabst et al., 2008) that have the potential to produce highly accurate results but require high levels of computation that are awkward to implement in real time. Mass-spring-based systems have already been shown to work effectively within graphics-based animation and simulation (Terasawa, 1996), portraying fast and aesthetically accurate behaviour of cloth (Howlett & Hewitt, 1998; Provot, 1997; Volino, Thalmann, Jianhua, & Thalmann, 1996) and other geometric objects (Theisel & Kreuseler, 1998). Most active research in sheet simulation has been restricted to the simulation of cloth-based applications (Eberhardt & Weber, 1999; Stewart, 1999; Baraff & Witkin, 1998). The system described here is composed of a number of adaptable three-dimensional sheet models that are inserted into a virtual environment containing several environmental parameters. These include adjustable levels of gravity (vertical downward force) and simulated heat-like thermal softening. Sheet parameters include variable levels of elasticity, plasticity, and moment stiffness. A variety of polyhedral objects may be “inserted” into the pseudo-immersive environment to demonstrate the effects of collisions of the sheet with three-dimensional objects. These objects also serve to highlight the behaviour of the sheet as changes in sheet plasticity/elasticity occur under a variety of temperature scenarios.

Even though it forms the principal focus of this study, the outcome of this work is not constrained solely to the purposes of modelling laminar three-dimensional objects.

By implementing a stacking system to the sheet structure, the numerical calculation engines presented here could be enhanced with the capability of modelling both two- or three-dimensional shapes, with examples including human body organs, animal skin and veins and other tubular structures such as synthetic electrical cables. The capabilities of the system would be expanded if augmented with haptic force feedback technology, thus allowing an operator to control the behaviour of the sheet, or three-dimensional model.

Typical applications are training simulations, aiding work/training in hostile environments, film animation, computer game development and medical imaging.

2.8 Summary

This chapter introduced the fundamental aspects of dynamic based simulation and the engineering required to represent the behaviour of cloth and laminar materials. Having examined the fundamental equations that drive cloth behaviour, the Kawabata reference method of evaluating elastic properties of cloth through experimental observation is described before detailing the use of dynamic simulation, particle systems and deformable simulation when modelling environments and synthetic materials. Techniques such as Finite Element Modelling, Mass-Spring systems, Integration Methods are evaluated before examining the methods of modelling cloth like materials.

3 Assessing the Theory

3.1 Introduction

In any study involving objects in the real world, environmental parameters are always incorporated to ensure credible behaviour. For example, if a tennis ball is held one metre above a smooth, flat wooden surface, it is known that on release the ball will fall until it hits the surface before bouncing back up, not achieving its initial altitude. If the ball is allowed to continue to fall back to the surface it will gradually reduce height with every successive bounce.

When such an environment is modelled in computer graphics, one starts off with a ‘clean sheet of paper’. There are no inherent physical rules to this computational world, no assumptions about the way that an object behaves, no gravity, no materials (and thus assumptions about their underlying properties such as strength, elasticity or mass), no friction, not even collision detection to prevent one object from passing through another. Instead, all these features must be programmed into the computer-based world that has been created.

To test the hypothesis that a multi-layer mass-spring system could provide credible simulations of stiff sheets, a test-bed application was implemented, referred to as ‘Aurora’. The Aurora system is comprised of a numerical simulation engine that computes the behaviour of the mass-spring sheet system using first, second and forth order ordinary differential equations (ODEs). This engine is called the Mass-Spring Simulation Engine (or MaSSE). An additional component of the Aurora system is the visual front-end that is used to control the simulation using a ‘World on Window’ (WoW) interface.

A simulation, such as the one described in this dissertation, must use a series of fundamental physical laws that can quickly and efficiently utilise the resources of the computer, without being hindered modelling all the complex behavioural aspects of the target subject (such as humidity, which is ignored). This balancing of computational speed and efficiency against the sufficient Laws of Physics to provide an effective visual replica of real-world behaviour forms a central element to the presented research.

The purpose of this chapter is to outline the fundamental physical laws and numerical methods that are used in the proposed system to model the sheet behaviour, before validating the results of a typical sheet modulus using these principal equations to verify that the model is based upon sound engineering principles.

The chapter then discusses the implementation of distinctive three-dimensional structures to represent bending moment in laminar objects prior to discussing the use of the three calculation engines in the model and detailing the limitations of using such approaches. These include model energy build-up, harmonics, numerical instability and sheet smoothing techniques for the resulting model behaviour.

3.2 Evaluation of Sheet Generation from Real World Parameters

To produce a realistic representation of a real-world sheet material into the simulation, a process is required that will allow input of the mechanical properties of a sheet material so they become parameters of the simulation model. This process involves the derivation of the sheet structures including spring stiffness, lengths and settings and is illustrated schematically in figure 3.1.

The simulation model requires values for the parameters listed below.

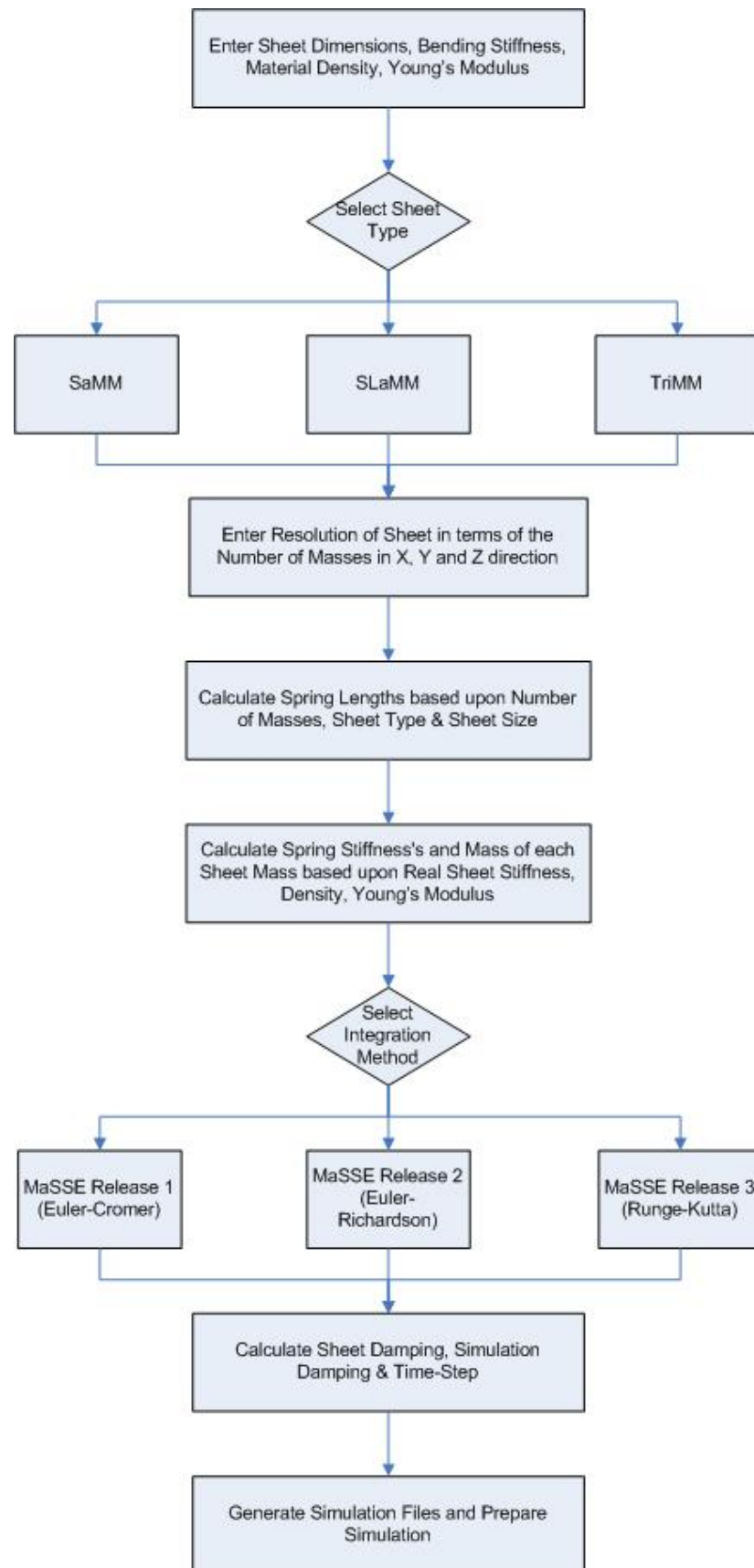
1. Bending Stiffness (determined using the Kawabata Evaluation System).
2. Young's Modulus (evaluated using the Kawabata Evaluation System).
3. Material Density (calculated by dividing the volume of the sheet by its mass).

These parameters can be measured using a number of different methods, including testing using the Kawabata Evaluation System that is described in section 2.5.2.

These values are then combined with the sheet dimensions and the number masses that will form the virtual sheet in each plane to produce the spring lengths and mass positions. Mass values for each *Mass* are determined by dividing the sheet mass by the number of masses comprising the sheet. Spring stiffness values for the inter-plane and inter-sheet springs and the are then calculated using the values for Young's modulus and Bending stiffness, taking into consideration the distance between the two virtual sheet layers (i.e. value h in Young's modulus, see section 3.3.3) and the distances between the interconnected masses.

Having selected the desired integration method, the simulation time-step and levels of sheet and simulation damping can be evaluated. The damping levels will be determined by the complexity of the integration method chosen and the time-step size, attempting to strike the balance between simulation performance and stability.

Figure 3.1 – Overview of Sheet Generation from Real-World Properties



3.3 Validation of the Simulated Material

3.3.1 Overview

To accurately evaluate the sheet dynamics engine and determine its effectiveness at representing the real properties of sheet materials comprised of plastic, the modelling engine and sheet structures must be mathematically correlated to its real world counterpart. Although the principal aim of this work is to evaluate the aesthetic effectiveness of the system at modelling sheet behaviour, the underlying mechanics should also be investigated. To achieve this, the relationship between the sheet structure and engineering principles must be evaluated using the underlying laws that encompass material forces and deformation, namely Hooke's Law and Young's Modulus of Elasticity.

As described in innumerable engineering and material science texts (Callister, 1997; Smith, 1993) the degree to which a structure deforms or strains is dependent upon the magnitude of an imposed stress. Hooke's law expresses stress as being equal to strain multiplied by the material's modulus of elasticity (otherwise known as *Young's Modulus*).

3.3.2 Hooke's Law

Hooke's Law states that within the elastic limit, the strain produced upon an object is directly proportional to the stress producing it.

Put another way, it states that

$$\sigma = E\varepsilon \quad \text{(Equation 3.1)}$$

Where σ = stress, ε = strain and E = Young's modulus.

The use of Hooke's Law is examined further in section 3.5.

3.3.3 Young's Modulus of Elasticity

The Young's modulus (or modulus of elasticity) is seen to represent a material's stiffness, or resistance to elastic deformation. The greater the modulus, the stiffer the material, or the smaller the elastic strain that results from the application of a given stress. As elastic deformation is non permanent, once the load is released, the material will reform into its original shape.

The value for Young's Modulus E is derived from

$$\frac{\text{tensile stress}}{\text{strain}} = \frac{P/A}{dL/L} \quad (\text{Equation 3.2})$$

Where

P is the applied force, A is the area upon which the force is applied, dL is the extension in length of the system, L is the initial length of the system, K is the spring constant,

$P = Kx$ where x is the compression or extension length.

Hence

$$E = \frac{PL}{dLA} = \frac{P}{A} * \frac{L}{dL} = \frac{kxL}{xA} = \frac{kL}{A} \quad (\text{Equation 3.3})$$

Spring stiffness, K (derived from Hooke's Law)

$$= \frac{\text{Applied Force}}{\text{Change In Length}} = \frac{P}{dL} \quad (\text{Equation 3.4})$$

Thus

$$K = \frac{AE}{L} \quad (\text{Equation 3.5})$$

with E = Young's modulus of elasticity.

For the purposes of the mass and spring model

$$A_{zz} = wh \quad (\text{Equation 3.6a})$$

Where w = distance between masses in a single layer, h = perpendicular distance between two layers (or the inter-sheet thickness)

And

$$A_{xx} = wl \quad (\text{Equation 3.6b})$$

Where w = distance between masses in a single layer, l = perpendicular distance between two layers (or the inter-sheet thickness)

Hence

$$K_{zz} = \frac{whE}{l} \quad (\text{Equation 3.7a})$$

$$K_{xx} = \frac{wlE}{h} \quad (\text{Equation 3.7b})$$

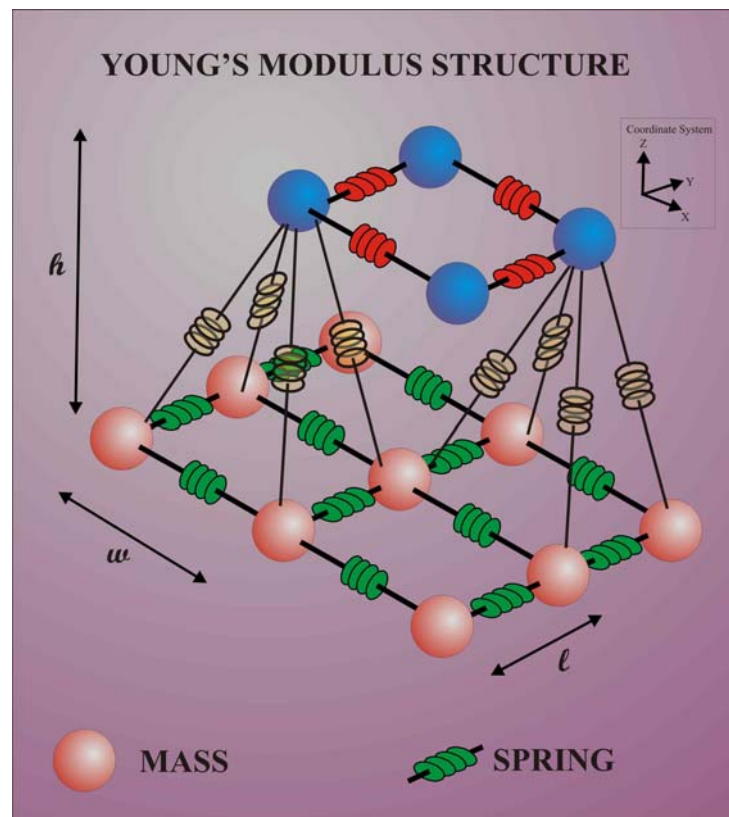
And

$$E_{zz} = \frac{Kl}{wh} \quad (\text{Equation 3.8a})$$

$$E_{xx} = \frac{Kh}{wl} \quad (\text{Equation 3.8b})$$

These values are illustrated for a two-layered example of a sheet structure in figure 3.2.

Figure 3.2 – Young's Modulus Structure



If $A = wh$, then

$$K_{zz} = \frac{EA}{l} = E_{zz} \frac{wh}{l} \quad (\text{Equation 3.9a})$$

$$E_{zz} = \frac{lK}{wh} \quad (\text{Equation 3.9b})$$

If $A = wl$, then

$$K_{xx} = \frac{EA}{h} = E_{xx} \frac{wl}{h} \quad (\text{Equation 3.10a})$$

$$E_{xx} = \frac{hK}{wl} \quad (\text{Equation 3.10b})$$

Thus

$$K_{xx} = \Sigma K \cos \theta \quad (\text{Equation 3.11})$$

Based upon these equations, it is possible to not only evaluate the modulus of Elasticity of any simulated three-dimensional sheet, but also to produce sheets that, based upon their dimensions and spring stiffness values have a specified Young's modulus.

3.3.4 Validation of the Sheet Modulus

Table 3.1 represents the parameters values used in two of the sheets that were used in the HCI user evaluation.

<i>Table 3.1 – Sheet Values For SLaMM HCI Test Sheets</i>		
Property	Euler-Richardson Sheet	Runge-Kutta Sheet
Sheet Dimensions	11 x 2 x 11	11 x 2 x 11
Sheet Area (x, y, z)	-1.25 – 1.25, 0.25 – 0.35, -1.25 -1.25	-1.25 – 1.25, 0.25 – 0.35, -1.25 -1.25
Spring Stiffness, K (N/M x 10 ⁻³)	205	405
<i>h</i> (M x 10 ⁻²)	0.10	0.10
<i>l</i> (M x 10 ⁻²)	0.25	0.25
<i>w</i> (M x 10 ⁻²)	0.25	0.25

For the Euler-Richardson sheet,

Strain Calculations (z direction)

$$lK = 0.25 \times 205 = 51.25 \text{ (M x } 10^{-2}\text{)}$$

$$wh = 0.25 \times 0.10 = 0.025 \text{ (M x } 10^{-2}\text{)}$$

$$E_{zz} = \frac{lK}{wh} = 2050 \text{ psi}$$

Strain Calculations (x direction)

$$hK = 0.10 \times 205 = 20.5 \text{ (M x } 10^{-2}\text{)}$$

$$wl = 0.25 \times 0.25 = 0.0625 \text{ (M x } 10^{-2}\text{)}$$

$$E_{xx} = \frac{hK}{wl} = 328 \text{ psi}$$

For the Runge-Kutta sheet,

Strain Calculations (z direction)

$$lK = 0.25 \times 405 = 101.25 \text{ (M} \times 10^{-2}\text{)}$$

$$wh = 0.25 \times 0.10 = 0.025 \text{ (M} \times 10^{-2}\text{)}$$

$$E_{zz} = \frac{lK}{wh} = 4050 \text{ psi}$$

Strain Calculations (x direction)

$$hK = 0.10 \times 405 = 40.5 \text{ (M} \times 10^{-2}\text{)}$$

$$wl = 0.25 \times 0.25 = 0.0625 \text{ (M} \times 10^{-2}\text{)}$$

$$E_{xx} = \frac{hK}{wl} = 648 \text{ psi}$$

Based upon these values and Equation (3.49), the Young's modulus of the Euler-Richardson sheet in the z direction is 2050 psi, with the Runge-Kutta sheet having a Young's modulus value (in the z direction) of 4050 psi. The Young's modulus of the Euler-Richardson sheet in the x direction is 328 psi, with the Runge-Kutta sheet having a Young's modulus value (in the x direction) of 648 psi.

Table 3.2 presents a list of Young's modulus for three common materials that are extracted from oil.

<i>Table 3.2 – Typical Material Values For Young's Modulus</i>		
Material	Young's modulus (E) in GPa	Young's modulus (E) in lbf/in² (psi)
Rubber (small strain)	0.01-0.1	1,500-15,000
Low density polyethylene	0.2	30,000
Polypropylene	1.5-2	217,000-290,000

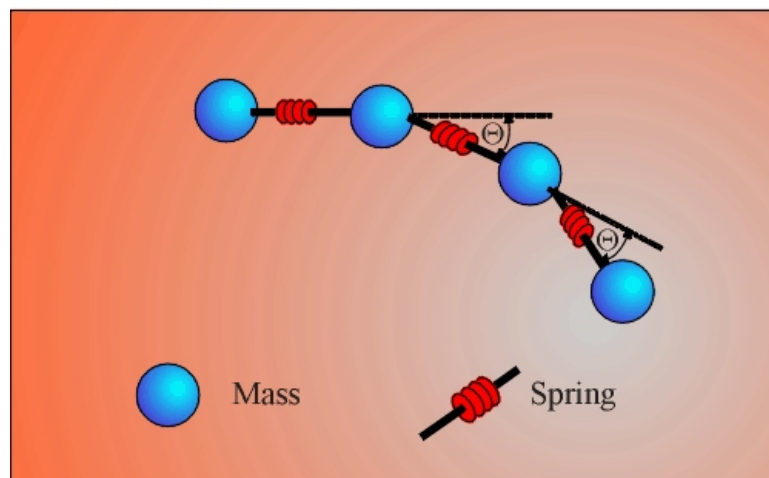
3.4 Bending Moment Representation

One of the novel aspects of the MaSSE system compared to traditional mass-spring systems is the unique way it deals with bending moment stiffness.

As previously discussed in section 2.7, the vast majority of mass-spring based laminar research employs two-dimensional structures (as illustrated in figure 3.2). Computationally expensive calculations are used to limit rotational stiffness in order to simulate bending moment. These methods are required as the methods also utilise implicit integration methods that, while resulting in enhanced levels of simulation stability also undermines the ability to reproduce bending moment. However this short-fall only represents a limited impact upon the most commonly simulated laminar material, i.e. cloth.

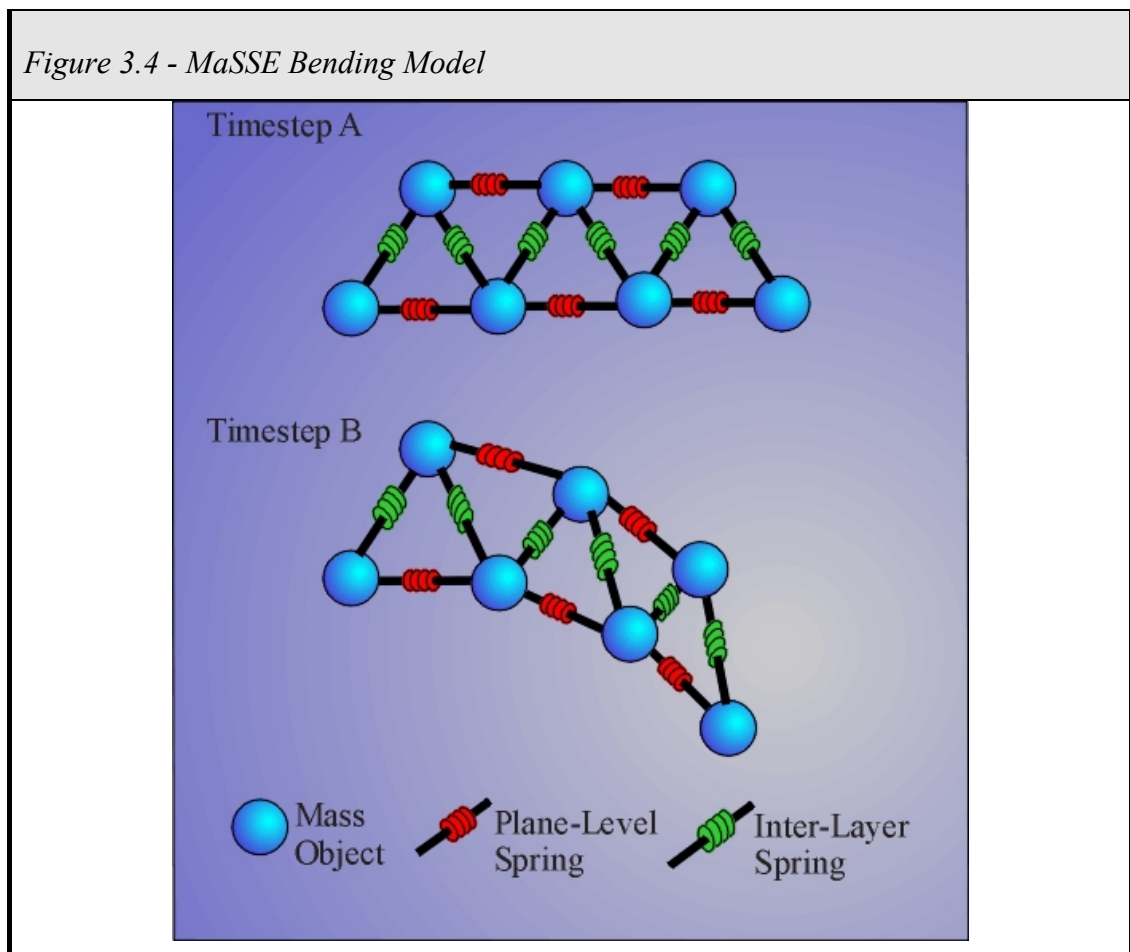
Rotational stiffness is achieved by implanting each mass with a horizontal axis component that identifies a horizontal plane. Then, through a set of calculations (that are performed in every simulation time-step), a comparison is made between the angles of the two opposite-end springs that are connected to each mass. If their angle value (denoted by θ in figure 3.3) is greater than arbitrary preset values then restrictions are placed on the angle of the *second* spring to ensure that θ remains at the preset rotational value.

Figure 3.3 - Rotational Stiffness Bending Model



Volino and Magnenat-Thalmann (2006) describe how bending stiffness is traditionally only applied using forces such as those provided by additional springs (for example Provot's flexion spring model (1997)) or through out-of-plane forces along surface normals (such as that presented in figure 3.2). The disadvantage of the former is that they can produce inaccurate results while the latter requires significant computation and is thus unsuitable for real-time or interactive environments. They then demonstrate how they have achieved bending stiffness for their cloth simulation using a "bending vector" to represent the surface bending angle.

Three-dimensional structures (see figure 3.4) are employed to represent the moment force through the use of interlayer springs that are connected to a second mass-spring layer. Together with experimentation in various sheet structures these lead to an efficient alternative to current approaches.



The work of Provot (1997) (see figure 2.9) illustrates an effective approach that uses 2-dimensional sheet structures. To allow for moment forces in his model, special ‘flexion’ springs are appended to this ‘conventional’ sheet structure. Flexion springs operate as an independent layer that overlays this structural mass-spring arrangement (as described in section 2.5.1). They not only have differing properties to these conventional springs (such as spring-stiffness), but they are also linked to every other spring in the sheet structure, thus helping to cater for an additional degree of angular stiffness. This use of a third dimension (depth) allows an overall greater level of flexibility, because the interlayer cross-springs can then be modified to cater for different levels of modular stiffness.

3.5 The Aurora Model Dynamics

Three different implementations (or releases) of the proprietary Mass-Spring Simulation Engine (MaSSE) were produced that are known as the MaSSE Release 1 (First Order Euler-Cromer), MaSSE Release 2 (Second Order Euler-Richardson) and MaSSE Release 3 (Fourth Order Runge-Kutta) engines, with each arrangement being adapted to suit its respective Ordinary Differential Equation (ODE) engine method. Each engine uses explicit Euler integration to determine the positioning of the masses after each time-step. Regardless of the depth of ODE calculation, the core approach is used to calculate all dynamics-based equations in the deformation environment. This design of the engine is easily extendable so that in future the MaSSE system (See Chapter 7) could cater for a diversity of additional shape simulations such as non-lamina models.

All of the methods are derived from Euler's method for solving Newton's laws of motion which define a particle (or in our case a *mass*) as an idealised object without internal structure.

Assuming $y(t)$ = position, $v(t)$ = velocity and $a(t)$ = acceleration then

$$v(t) = \frac{dy(t)}{dt} \quad (\text{Equation 3.12a})$$

$$a(t) = \frac{dv(t)}{dt} \quad (\text{Equation 3.12b})$$

A particle's acceleration is based upon the net force acting upon it, which in our case is a result of gravity and the push and pull based by a mass's interconnecting springs. Therefore when F = net force and m = internal mass, then

$$a(t) = \frac{F(y, v, t)}{m} \quad (\text{Equation 3.13})$$

This simply states that force is dependent upon position, velocity and time.

The motion of a particle requires the solution of two coupled-first order differential equations (Equations (3.12a) and (3.13)) that are combined to form a second order differential equation for determining the new position. i.e.

$$\frac{d^2 y(t)}{dt^2} = \frac{F}{m} \quad (\text{Equation 3.14})$$

Where g = gravitational (downward) acceleration, y_0 = initial displacement and v_0 is the initial velocity, then Equation (3.14) can be rewritten as:

$$V(t) = V_0 - gt \quad (\text{Equation 3.15})$$

and

$$y(t) = y_0 + v_0 t - \frac{1}{2} g t^2 \quad (\text{Equation 3.16})$$

Euler's method for solving Newton's laws of motion is generalised into:

$$v_{n+1} = v_n + a_n \Delta t \quad (\text{Equation 3.17a})$$

and

$$y_{n+1} = y_n + v_n \Delta t \quad (\text{Equation 3.17b})$$

3.5.1 Sheet Calculation Engines

MaSSE Release 1 - The Euler-Cromer First Order Engine

MaSSE Release 1 (figure 3.5) was designed for first-order calculations which, due to its simple formatting offered the fastest and most computationally cost effective system.

The Euler-Cromer method for solving Newton's laws of motion is based upon Euler's original equations (see equations (3.18) and (3.19)) although it differs by computing velocity at the end of the interval, as opposed to the beginning of the time-step.

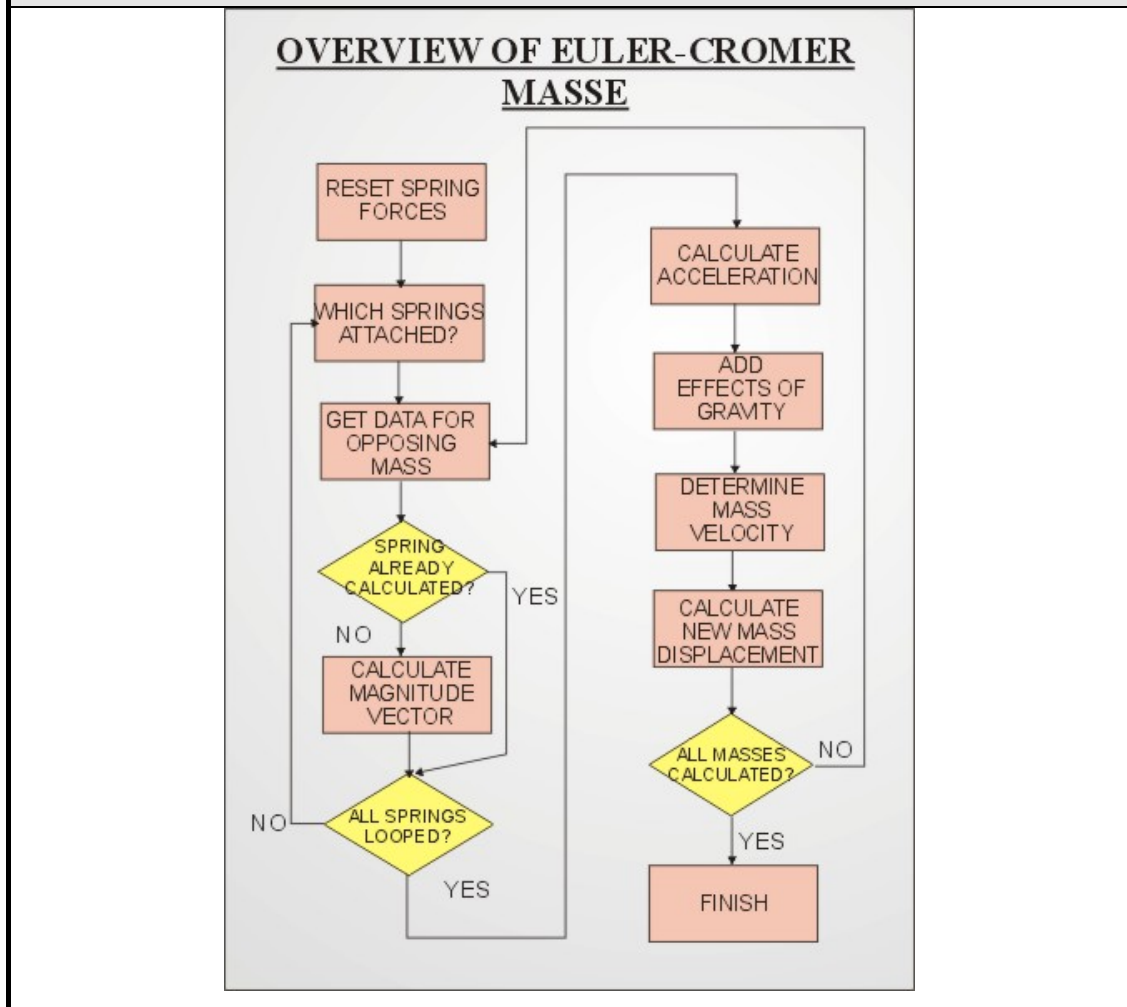
$$v_{n+1} = v_n + a_n \Delta t \quad (\text{Equation 3.18})$$

and

$$y_{n+1} = y_n + v_{n+1} \Delta t \quad (\text{Equation 3.19})$$

Although this is only a minor difference between the Euler and Euler-Cromer approaches, there was slim improvement in system stability through use of the Euler-Cromer approach. This reinforces the work of Gould and Tobochnik (1996) in their book documenting computer simulation techniques.

Figure 3.5- MaSSE Release 1 Dynamics Flowchart Activity Diagram



MaSSE Release 2 - The Euler-Richardson Second-Order Engine

MaSSE Release 2, a mid-point approach forms the best solution over the Euler and Euler-Cromer methods as it is able to compute velocity during the middle of the interval, as opposed to the beginning or the end of the time-step. Although the Euler-Richardson mid-point approach requires an additional level of calculation to cater for the intermediate step between intervals, it is useful for velocity-dependent forces. The approach also does as well as other simple algorithms for forces that do not depend upon velocity (Gould and Tobochnik, 1996, p42). The overhead of this increased accuracy has been minimised in the engine design (figure 3.6) due the software's capability of calculating the mid-step as an extension of the full-step computation

without the need to retain any additional data over and above that required for the first-order system.

The method works on the assumption that $t_{mid} = t + \Delta t/2$. The algorithm then calculates the force, $F(y_{mid}, v_{mid}, t_{mid})$ and the acceleration a_{mid} at time t_{mid} . The new position x_{n+1} and velocity v_{n+1} at time t_{n+1} is found using v_{mid} and a_{mid} . The Euler-Richardson algorithm is summarised as:

$$a_n = \frac{F(y_n, v_n, t_n)}{m} \quad (\text{Equation 3.20})$$

$$v_{mid} = v_n + \frac{1}{2} a_n \Delta t \quad (\text{Equation 3.21})$$

$$y_{mid} = y_n + \frac{1}{2} v_n \Delta t \quad (\text{Equation 3.22})$$

$$a_{mid} = \frac{F(y_{mid}, v_{mid}, t + \frac{1}{2} \Delta t)}{m} \quad (\text{Equation 3.23})$$

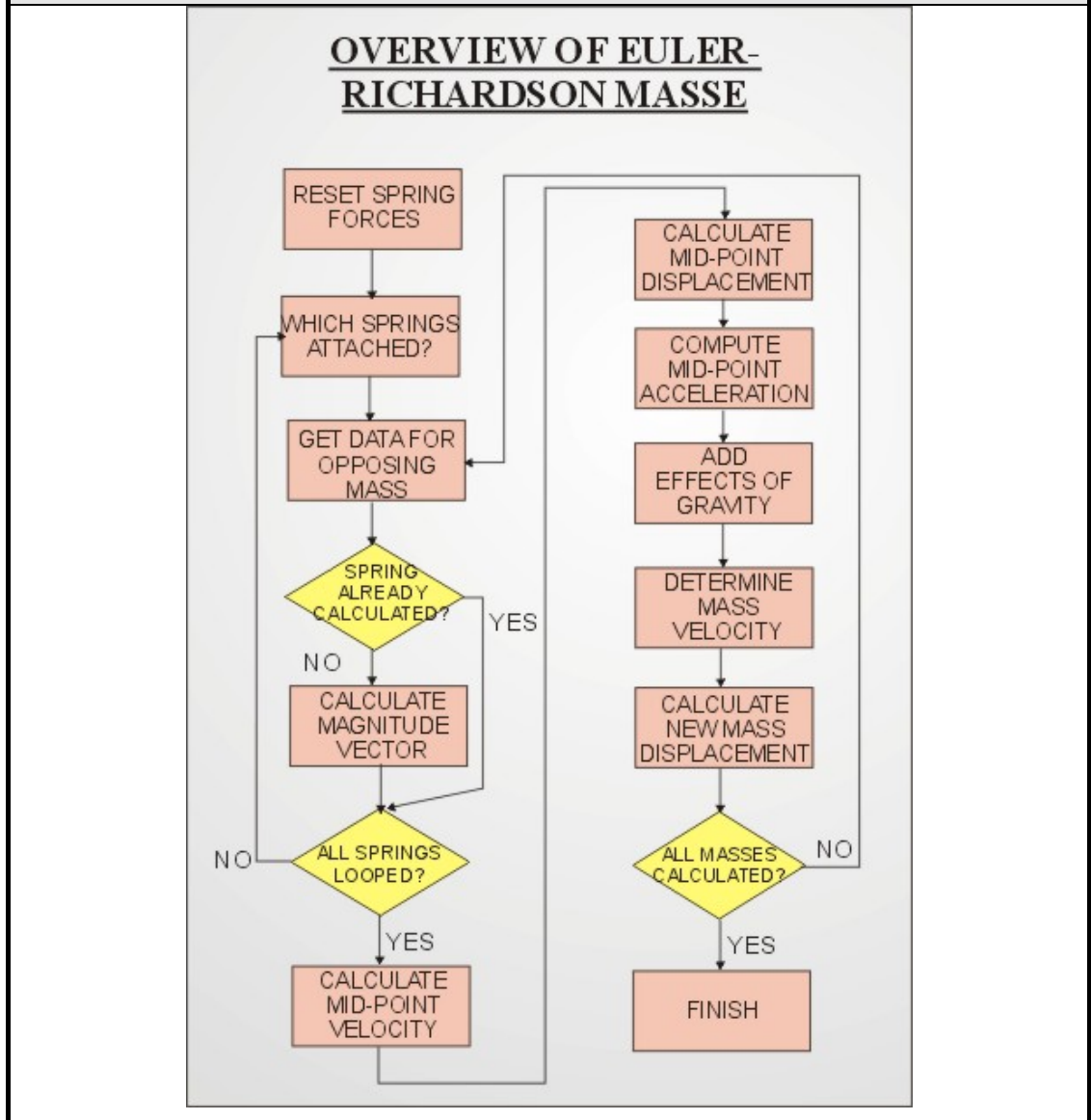
and

$$v_{n+1} = v_n + a_{mid} \Delta t \quad (\text{Equation 3.24})$$

$$y_{n+1} = y_n + v_{mid} \Delta t \quad (\text{Equation 3.25})$$

As previously mentioned, although the Euler-Richardson system requires double the number of calculations for each time-step, the algorithm is in fact significantly faster and more reliable during actual operation than the Euler-Cromer system since, due to enhanced levels of calculation accuracy of the second order method, larger time-steps with greater sheet sizes are now catered for.

Figure 3.6- MaSSE Release 2 Dynamics Flowchart Activity Diagram



MaSSE Release 3 - The Fourth-Order Runge-Kutta Engine

Lastly MaSSE Release 3 implemented the Runge-Kutta method readily extends to allow the numerical solution of simultaneous and higher order equations.

The general form of all Runge-Kutta methods is:

$$y_{i+1} = y_i + \sum_{j=0}^{r-1} w_j k_j \bigg/ \sum_{j=0}^{r-1} w_j \quad (\text{Equation 3.26})$$

where

w_j = constant weighting of the coefficients

r = order of the method

k_j = estimates of the change in the 'y' evaluated at 'r' locations within the range 'h'.

Fourth order Runge-Kutta is expressed as the following.

Consider acceleration = $\dot{v} = \ddot{y} = f(t, y, v)$

with velocity = $\dot{y} = v$

then

$$v_{n+1} = v_n + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) \quad (\text{Equation 3.27})$$

and

$$y_{n+1} = y_n + \frac{1}{6}(L_1 + 2L_2 + 2L_3 + L_4) \quad (\text{Equation 3.28})$$

Where velocity is derived by:

$$K_1 = h.f(t_n, y_n, v_n) \quad (\text{Equation 3.29a})$$

$$K_2 = h.f(t_n + \frac{h}{2}, y_n + \frac{L_1}{2}, v_n + \frac{K_1}{2}) \quad (\text{Equation 3.29b})$$

$$K_3 = h.f(t_n + \frac{h}{2}, y_n + \frac{L_2}{2}, v_n + \frac{K_2}{2}) \quad (\text{Equation 3.29c})$$

$$K_4 = h.f(t_n + h, y_n + L_3, v_n + K_3) \quad (\text{Equation 3.29d})$$

Displacement is derived by:

$$L_1 = h.v_n \quad (\text{Equation 3.30a})$$

$$L_2 = h.(v_n + \frac{K_1}{2}) \quad (\text{Equation 3.30b})$$

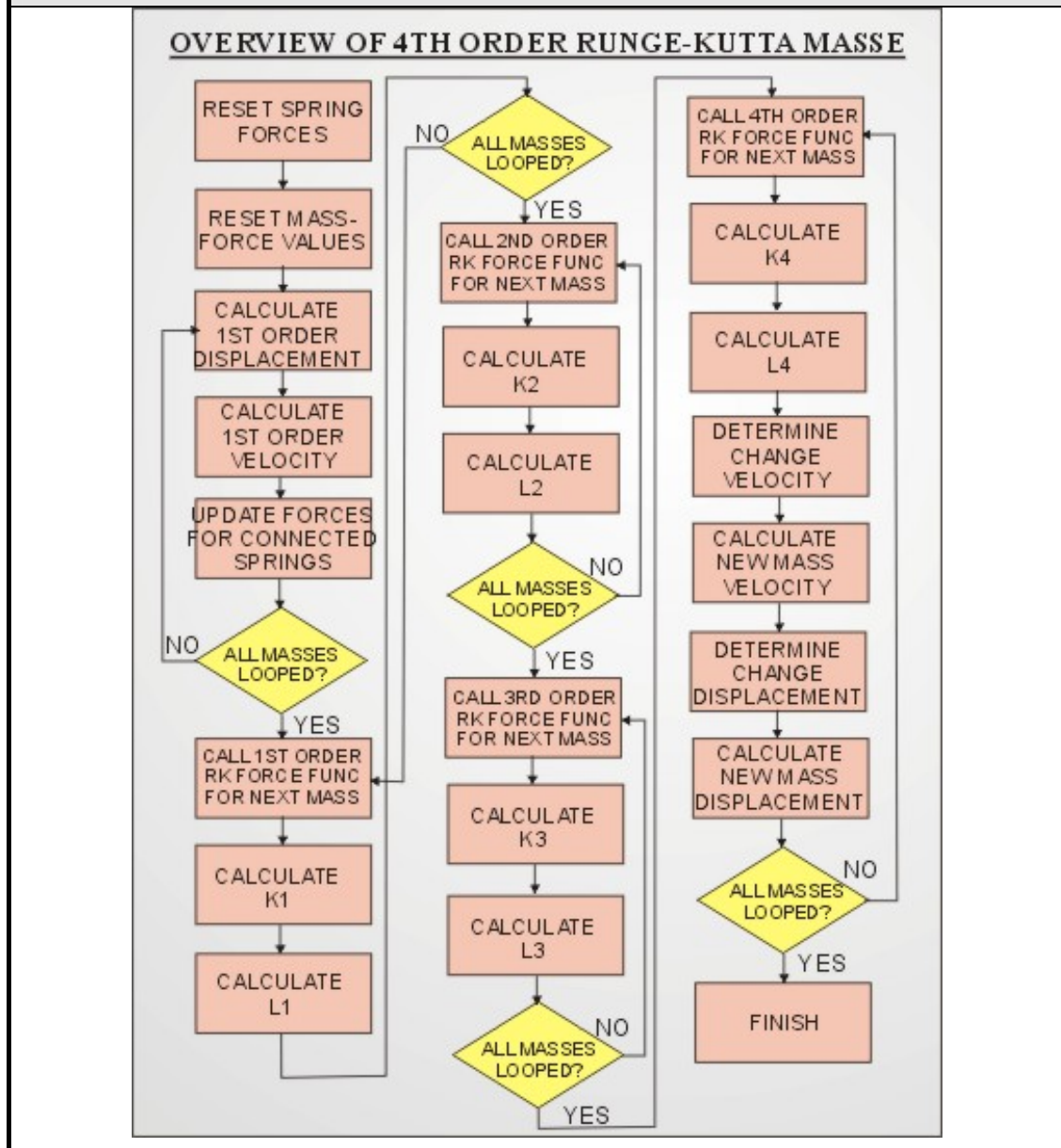
$$L_3 = h.(v_n + \frac{K_2}{2}) \quad (\text{Equation 3.30c})$$

$$L_4 = h.(v_n + K_3) \quad (\text{Equation 3.30d})$$

The final mutation of the MaSSE system (see figure 3.7) was developed for performing fourth-order calculations within the environment. Although computationally expensive, this approach is regarded as the common benchmark for most dynamics based calculation work (Volino et al. (2005); Gould and Tobochnik (1996)). Due to its complex nature, rather than the system dealing with each of the four equation levels individually for each *mass*, the system must predict the forces, velocities, accelerations

and positions for the entire sheet for each level, prior to calculating the next level of the differential equation. As one would expect, this not only produces a considerable computational burden upon the processing hardware, but more significantly requires the storage of large amounts of data for the entire sheet for every time-step.

Figure 3.7- MaSSE Release 3 Dynamics Flowchart Activity Diagram



The structure of the engine designs in relation to the depth of Ordinary Differential Equations (ODE) calculation is covered in greater depth later in the chapter.

This engine was designed and developed using a modular (object oriented) structure using Microsoft Visual C++ release 6. The environment also utilises elements of the Microsoft Foundation Classes, OpenGL, the Graphics Library Utility Toolkit (GLuT) and the Graphics Library Utility Interface (GLUi).

3.6 Physical Simulation

The ability to effectively model the behaviour of real world objects in a virtual environment requires a mixture of computationally expensive empirical techniques. Although they are capable of accurately modelling the real world, these numerical techniques do possess a limited scope of applications.

3.6.1 Damping Model

Spring and dashpot models are widely used to represent the viscoelastic modes of deformation within amorphous thermoplastic material (Higgins, 1994). In these cases it is assumed that the springs obey Hooke's Law whereby the strain (ε) is proportional to stress (σ) or:

$$\varepsilon = \frac{\sigma}{E} \quad (\text{Equation 3.31})$$

When damping Mass-Spring systems in three-dimensions, the damping force vector placed upon a spring is determined using:

$$\vec{F} = -K_d \frac{(\vec{V}_A - \vec{V}_B) \cdot \vec{L}}{|\vec{L}|} \frac{\vec{L}}{|\vec{L}|} \quad (\text{Equation 3.32})$$

where

K_d = Damping coefficient

A and B = the two masses connected by the spring

L = vector pointing from mass B to mass A

F = the damping force inverted on mass A

V_A and V_B = Velocities of masses A and B respectively, with the damping force opposing this motion.

A number of possibilities exist to represent the effects of stress upon the sheet structure. Figure 3.8 illustrates the time-strain curve for a simple spring with no damping effect. This indicates how once the force F is released, the strain levels fall immediately to zero.

Figure 3.8 – Time-Strain Curve for Spring

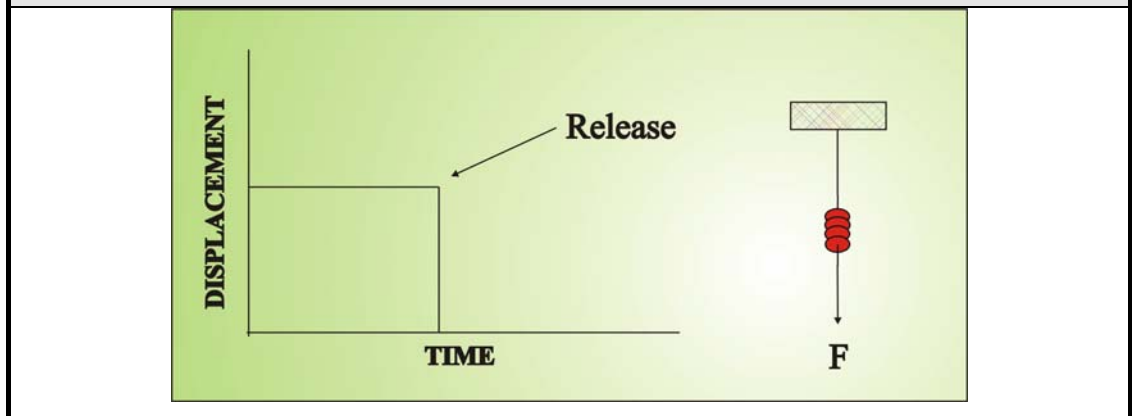


Figure 3.9 portrays the strain behaviour of a system with no spring elements but instead comprising simply of a dashpot damper. Whilst encountered the force, the strain increases proportionally however once F is removed, the level of strain remains constant.

Figure 3.9 – Time-Strain Curve for Dashpot Damper

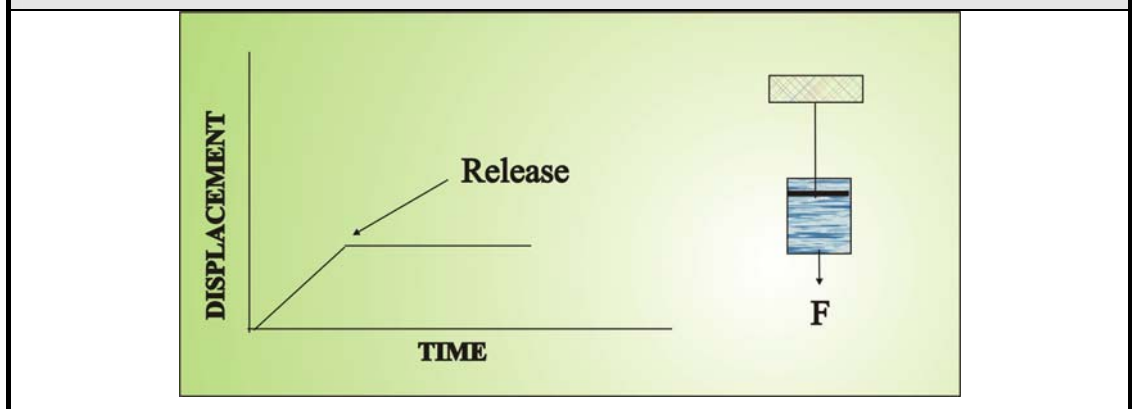


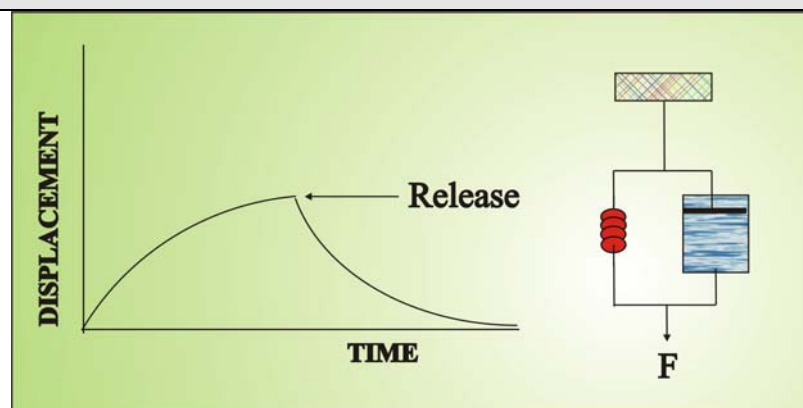
Figure 3.10 illustrates the behaviour of a spring and damper system operating in series whereby the level of strain grows proportionally until F is removed, whereby the level of strain falls to a predetermined level, upon which it remains constant.

Figure 3.10 – Time-Strain Curve for Spring and Damper (Maxwell Element)



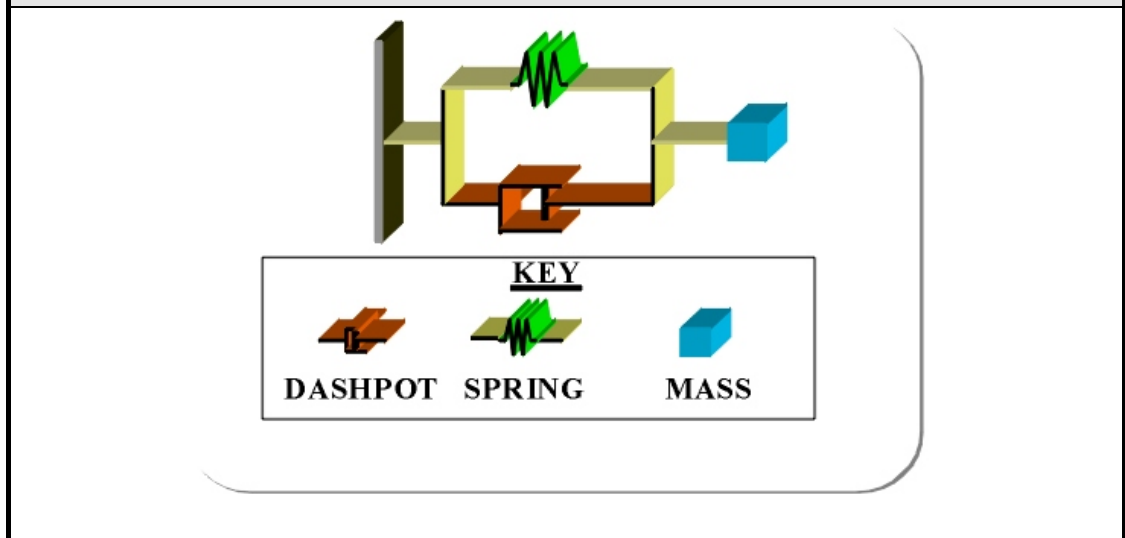
Figure 3.11 highlights the behaviour of a spring and dashpot model placed in parallel. This offers the benefit of a steady, smooth increase in strain until the force is released at which point the level of strain returns gradually to zero.

Figure 3.11 – Time-Strain Curve for Parallel Spring and Dashpot (Voigt-Kelvin Element)



For the modelled sheet simulation, a simple generalised single degree of freedom model is used to represent frictional forces acting upon the sheet within the environment (see figure 3.12).

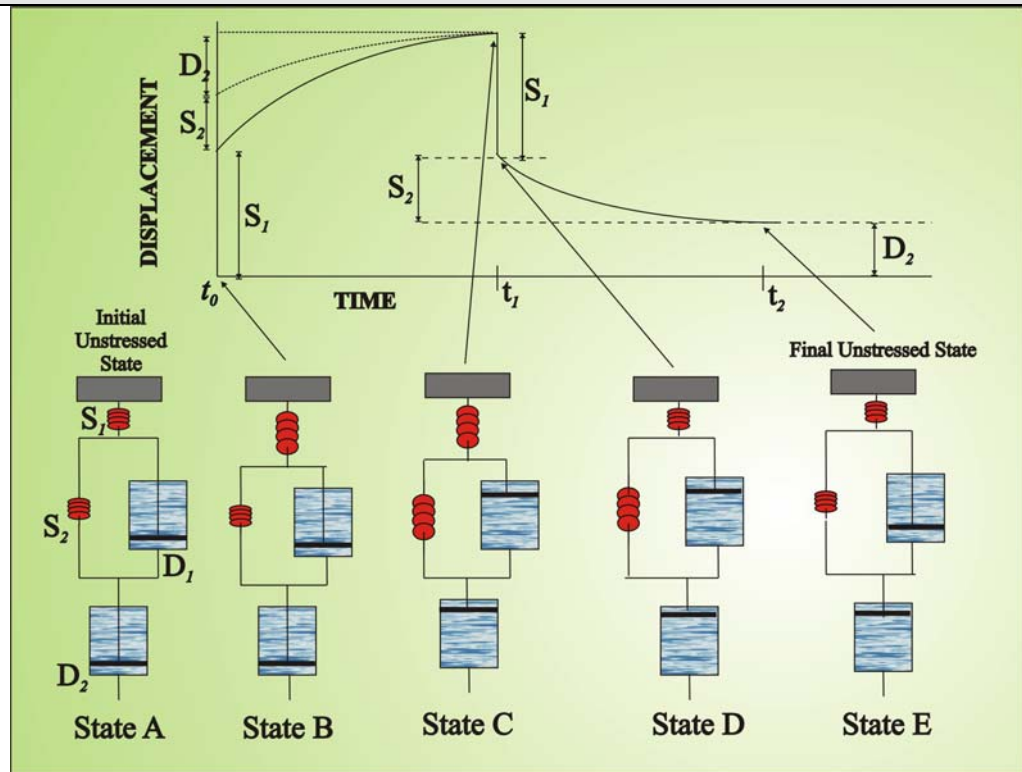
Figure 3.12 - Simple Damping Model



This method is considered to be the established approach to damping in spring-based systems (Rao, 1990). The benefit of such a simple model is the ability to extract energy from the sheet model (to represent the effects of friction, heat, sound etc.) with minimal computational overhead.

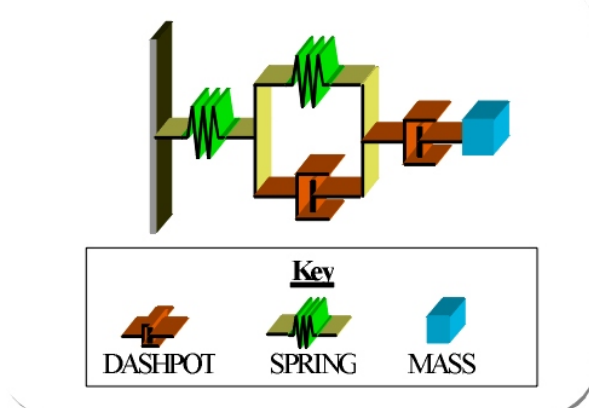
Based upon the behaviour of figures 3.8-3.11, it is possible to derive the strain/time relationship for more complex systems such as the one illustrated in figure 3.13. If the initial system state (marked 'State A') represents the unstressed state, then the sudden application of force results in the elastic deformation S_1 (at time t_0), indicated by 'State B', which is then followed by the viscoelastic response S_2 in the S_2/D_1 unit, coupled with viscous flow in the dashpot D_2 at time t_1 (as shown in State C). In State D, the displacing force is removed and the elastic element S_1 relaxes immediately, before the viscoelastic element (in State E) also relaxes during the period t_1 - t_2 , however the viscous flow D_2 is never recovered.

Figure 3.13 – Strain curves for the Advanced Damping System (Higgins, 1994)



Based upon these findings, when investigating a tuned damping model for the specific research requirements, a structure for representing viscoelastic sheet behaviour was proposed that provided greater levels of realism (see figure 3.14 below).

Figure 3.14 - Advanced Damping Model



This is an extension of the Voigt-Kelvin element spring and dashpot model, which has been combined with a Maxwell element system to produce what is expected to be an efficacious model for simulating viscoelastic behaviour. Due to the extensive computational overhead of implementing the method through the use of Lagrange's equations, the decision was made not to adopt this approach in the current system. This is covered in further depth in the Discussion section of Chapter 7.

3.6.2 Gravity

The World on Window (WoW) environment is equipped with a gravity force to provide a realistic simulation of sheet behaviour whilst undergoing simulated deformation as the sheet is forced into contact with the polyhedra objects.

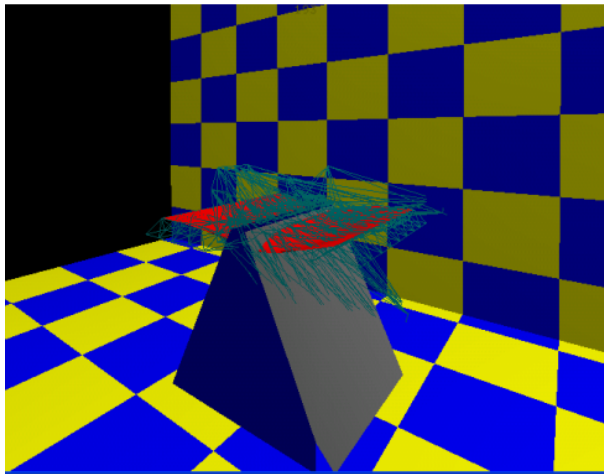
To enable a user to gain a clearer view of the collision and deformation process, the level of gravitational force has been deliberately reduced from 9.81 m/s^2 to one tenth of this value. This produces the effect of creating a smoother simulation with the sheet falling effortlessly from its initial position down to the surface. This change is particularly apparent when operating the simulation with increased step-sizes when the sheet would otherwise be viewed jumping from one frame to the next. It should be emphasised that the value of gravitational acceleration may be adjusted by the user at runtime.

3.6.3 Simulation Instability

Within every simulation time-step, each mass particle changes the impulse vector (v) for each neighbour. The new vector is generated from the sum of the old vector (from the previous time-step) combined with the additional influences that would influence its behaviour since the preceding time-step (e.g. gravity etc.). When pooled with the certainty that the particle is also influenced by its previous direction, any sudden change of direction can lead to long term consequences for the numerical stability of the model.

A principal issue in many simulations, particularly those that adopt explicit integration techniques are that energy gains may eventually cause a simulation to become unstable and “explode”. This has various implications depending upon the type of simulation being portrayed, varying from erratic behaviour of characters that use integration to drive kinematics with the characters ‘flying’ wildly around the scene, through to the physical “explosion” of mass-spring systems with masses (and their interconnecting springs) flying off in random directions at high velocity (see figure 3.15). This is a well known issue in any simulation, but particularly in mass-spring systems (Plath, 2000) as the distance between masses becomes too large causing the springs to become overextended and the sheet appearing to “explode”.

Figure 3.15 – A Demonstration of Sheet Instability



The level of simulation stability will hinge upon a number of factors that include:

- a) The effectiveness of the simulation algorithm to prevent energy build-up. This may include the use of implicit integration (with its shortcomings that include a heavy computational overhead) or controlled explicit integration, which while more efficient, will require smaller time-steps to maintain adequate levels of stability.
- b) The simulation time-step, with smaller time-steps enhancing stability, but at the cost of real-time performance and the amount of computation required.
- c) The order of the integration with higher order approaches enhancing stability, but at the expense of performance and computation. Fourth order integration (or even fifth

order with error correction) provides the maximum level of stability but can involve punishing levels of computation. This can be off-set to some degree by being able to safely increase the simulation time-step size, or through the use of dynamic time-steps (through the use of error correction) to provide a faster simulation whilst maintaining a specified level of stability. Carefully calibrated 2nd order integration approaches (such as the mid-point technique) can provide suitable levels of performance and stability whilst avoiding the computational overhead.

- d) The level of damping attributed to the simulation. Increased levels of damping increase system stability although over-damping will result in the sheet behaving as if it were numb and anesthetised. Baraff and Witkin (1998) provided a detailed comparison of damping methods in cloth simulation for their paper describing their implicit-based cloth simulation system.

An additional factor in retaining system stability is the use of multiple layers of sheet with additional layers of interconnected springs providing further support for the structure compared to traditional two-dimensional sheet models. This premise is discussed and evaluated later in the thesis in chapters 5 and 6 where the three-dimensional sheet structures are evaluated against their traditional two-dimensional counterparts before undergoing an aesthetic usability evaluation by a representative population sample.

3.7 Periodic Motion

Periodic motion is a well known problem in sheet simulation whereby each mass-spring arrangement acts a harmonic oscillator that results in harmonic motion.

When linking a series of masses and springs together the effects of harmonic motion and corresponding impact of periodic motion must be considered to prevent the system from becoming unstable, leading to simulation failure. Shearer et al. (2009) discusses the subject of a mass's planar motion when anchored by two springs in a Hamiltonian system.

This section shall briefly summarise vibration and harmonic oscillation with their corresponding affect on a mass-spring system.

3.7.1 Vibration

“Vibration is related to the cyclic variations of force and displacement, and as such is regarded as one of the most important aspects of dynamics” (Steidel, 1989).

Vibration is present in several forms:

Free Vibration: Occurring without externally applied forces, it arises when an elastic system is displaced or given some initial velocity (for example, from an impact).

Forced Vibration: Resulting from the application of external forces, forced vibrations can be periodic (i.e. repeating itself in regular intervals), aperiodic or random (i.e. at irregular intervals).

Both forms of vibration allow damping to correspond to the dissipation of energy. A second variation in vibration is the number of independent co-ordinates needed to describe complete motion, otherwise known as the number of degrees of freedom.

Linear vibration is described as the circumstances under which dissipative (damping) forces are proportional to the velocity of motion, the restoring forces are proportional to displacement and the internal forces are proportional to acceleration.

3.7.2 Harmonic Oscillation

For mechanical oscillation to occur, a system must possess two quantities, *elasticity* and *inertia*. When a system is displaced from its equilibrium position, the elasticity provides a restoring force such that the system tries to return to equilibrium. The inertia property causes the system to overshoot equilibrium. It is this constant interplay between the elastic and the inertia properties that allow oscillatory motion to occur. The natural frequency of oscillation is related to the elastic and inertia properties by

$$\omega_0 = 2\pi f_0 = \sqrt{\frac{\text{elasticity}}{\text{inertia}}} \quad (\text{Equation 3.33})$$

This forms one of the fundamental principles surrounding mass-spring networks. When one applies Newton's second law $F = ma$ to a mass, the equation is easily extended to represent an oscillating system with the constant k being used to represent the elastic (or spring stiffness) restoring force and m representing the inertia (or mass value) that causes the equilibrium overshoot.

This allows us to obtain the equation of motion for the system, which is calculated as

$$m \frac{d^2x}{dt^2} + kx = 0 \rightarrow \frac{d^2x}{dt^2} + \frac{k}{m}x = 0 \rightarrow \frac{d^2x}{dt^2} + \omega_0^2 x = 0 \quad (\text{Equation 3.34})$$

Where $\omega_0 = \sqrt{\frac{k}{m}}$ (radians/sec) is defined as the natural oscillating frequency.

Any vibration within the model presented here is harmonic as non-harmonic vibration is only caused when the material used to construct the spring does not follow Hooke's law.

3.8 Summary

This chapter has described the rationale behind the design of the MaSSE system that is used to investigate the proposition that stiff-sheet materials can be effectively modelled using multi-layer mass-spring systems.

The chapter introduced the engineering mechanics that underlie mass-spring systems (combined with harmonic motion), its application in modelling laminar models, the differences between existing two-dimensional approaches and the proposed three-dimensional technique.

The numerical calculation engines that are employed in the system dynamics engine are also described, together with the reasons for, and methods of system damping that are applicable and/or employed by the system.

The Chapter finally examines the issues of conducting numerical simulation (such as simulation instability) and how these issues are controlled and managed.

The implementation of the three-dimensional mass-spring system shall be described in the following chapter.

4 System Implementation

4.1 Introduction

This chapter will describe the approach taken to create the system, the reasoning behind the build decisions and their benefits. These choices included the organisation of classes, the selected sheet structures and the design rationale behind the World on Window (WoW) environment.

Like any structure, computer software can be developed using any one of a variety of design methodologies. The eventual choice is often dependent on the size and purpose of the application, the adherence of good programming practice particularly in relation to the programming language, and the coding practices of the developer. It is important that any complex prototype adopts a formal structure to aid the development process, allowing easy extendibility of the software through a modular approach and understanding. An example of good programming practice is to base a simulation model's functionality upon real-world, or in this case, engineering-based metaphors.

To aid in implementing this example of good programming practice into the described system, an object-oriented methodology and programming structure was applied to ensure that the final system was flexible (to easily cater for varying sheet sized), adaptable (to allow complex three-dimensional objects, including laminar and non-laminar models to be simulated) and logically structured (to aid further, future extendibility) and system understanding.

4.2 System Design

4.2.1 Design Space Analysis

Design Space Analysis (DSA) is the method of recording the decision process in System Design as it undergoes the maturing process. DSA allows designers to back track through design decisions to determine when mistakes were made, assist in their rectification and in the prevention of repeat occurrences.

The output of any DSA is not simply the final product but also the routes that were taken to reach that point.

4.2.2 Problems and Questions

The design problems within the sheet simulation system may be divided into four categories, these being

Pre-Simulation Data

These are concerned with the facilities to enter large amounts of data required to initiate a stable simulation solution. E.g. the simulation sheet structure, data on any polyhedra that are used in the simulation, initial values for sheet mass displacements, force values etc.

Data Storage

Storage questions are related to the storage of post simulation data. For example, post-simulation sheet-mass displacements etc.

Parameter Variables

These are the simulation variables that are either required before the simulation can start, and/or could be adjusted during the simulation itself. Examples are the simulation time-step, the method of numerical integration etc.

HCI Interface

These points focus upon the user interface itself, describing what permutations and options were considered and eventually selected. Examples are the form of smoothing algorithm and any other visual options (such as viewing the masses and/or springs) that are used in the simulation.

4.2.3 Questions within the Design Space

The following questions must be answered to gain a full perspective of the sheet simulation design space.

Pre-simulation data

1. What customisable data is required to initiate a sheet simulation?
2. How should this data be accessed?
3. Should an ability to modify this data exist?
4. Which applications should be used to access this data?

Data Storage

1. What data (if any) requires storage?
2. How should this data be stored?

Parameter Variables

1. Should data be allowed to be modified during program execution?
2. Which variables should be modifiable?
3. How should these variables be modified?

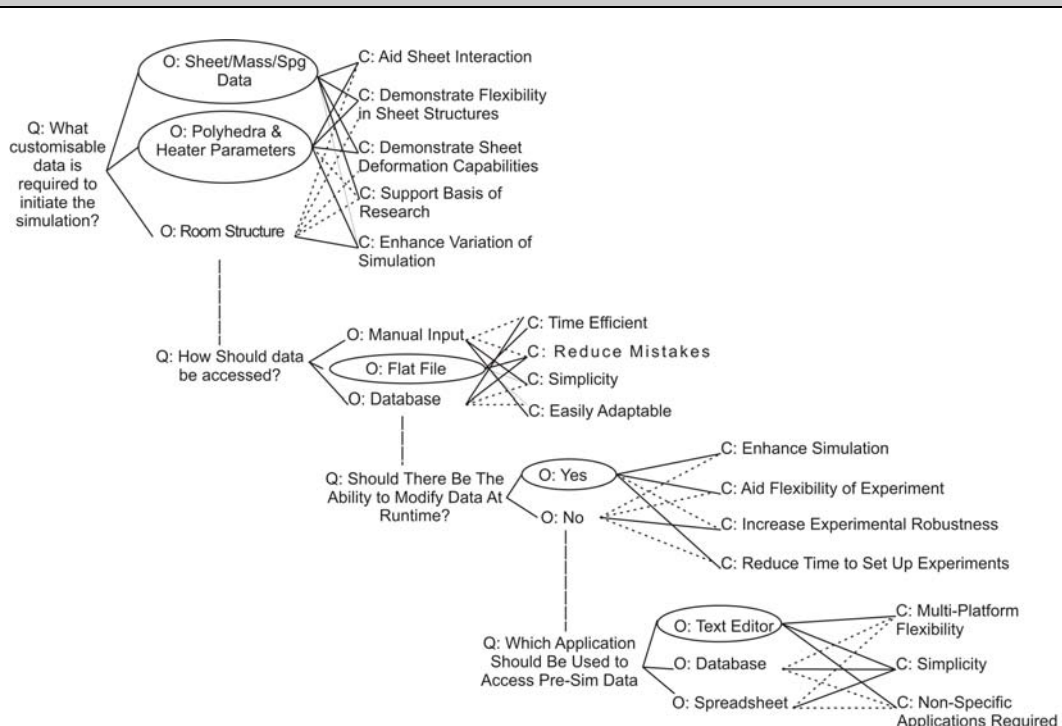
HCI Interface

1. Should the interface be adaptive or customisable?
2. What level of user interaction should the interface allow?
3. Are system short-cuts required?
4. Should short-cuts be screen, keyboard or mouse based?

4.2.4 Questions Options Criteria Diagram

Based upon the questions described in the previous section, it is possible to generate the Questions, Options, Criteria (QOC) diagrams which are used to demonstrate the design path undertaken by the system architect. Figure 4.1 describes the options and decision making process required to determine what data parameters, what access parameters for entering the data and what structure format to use to store the information for the pre-simulation phase of the simulation.

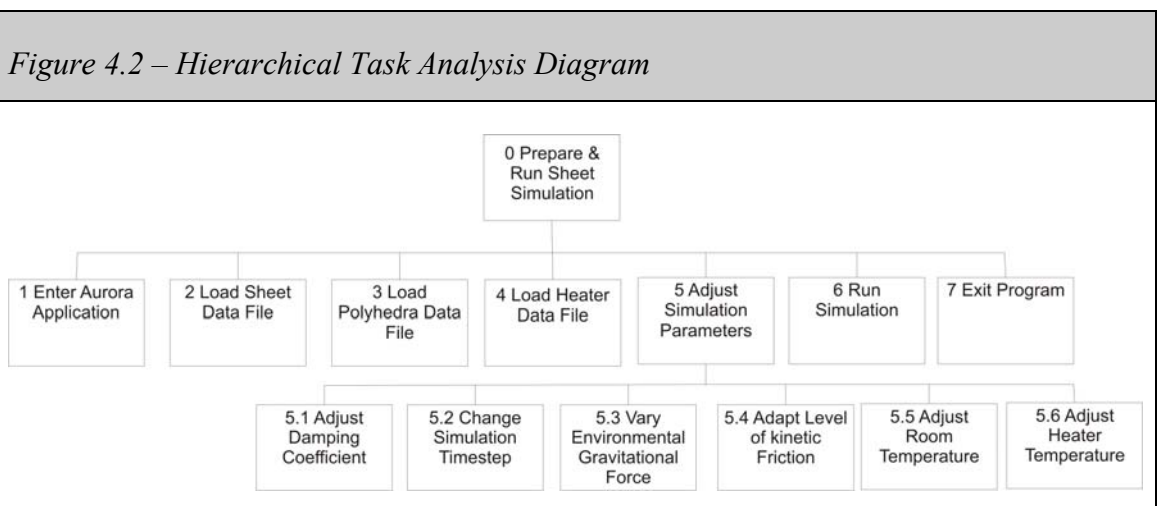
Figure 4.1 – Pre-Simulation QOC Diagram



4.2.5 Hierarchical Design Analysis

Hierarchical Design Analysis is used to map the interface design of a system to formalise a structure that allows a user to clearly determine the processes required to prepare and then run a simulation.

Figure 4.2 highlights the tasks that maybe required to prepare and execute the sheet simulation program, although through the use of default parameters not all actions require implementation with steps 4 and 5 (and their corresponding sub-steps) forming optional tasks.



4.2.6 Unified Meta language (UML)

The Unified Meta Language is a non proprietary specification language for object modelling. As a general purpose modelling language, UML includes standardised graphical notation that is used to create an abstract model of a system, otherwise known as the UML model. The model may then be used in the systems mapping and development phases of the software lifecycle.

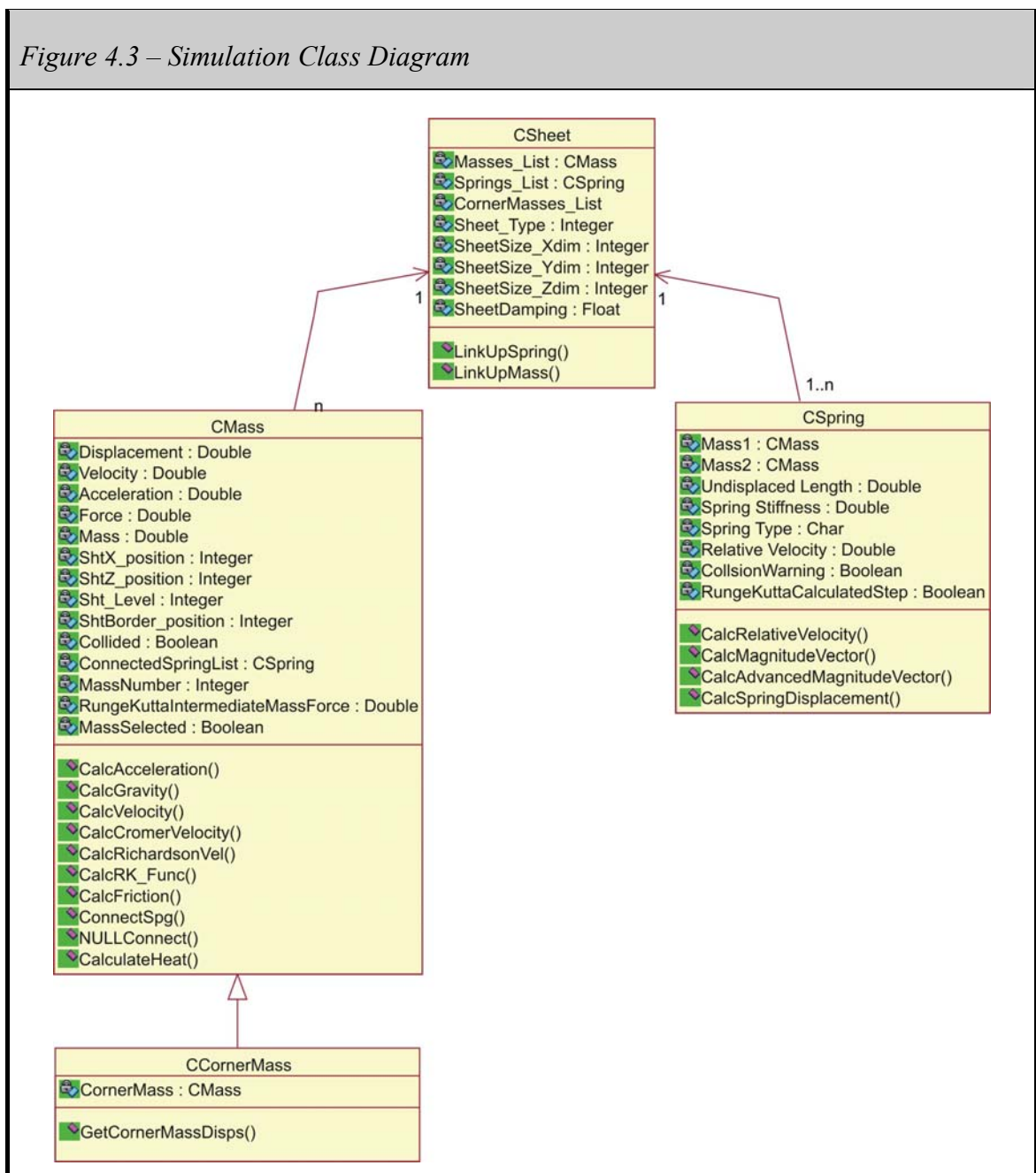
Examples of possible diagrams that can be used to produce the UML model include the Use Case Diagram, Class Diagram, Interaction Diagrams (including the Sequence and Collaboration Diagrams), State Charts, Activity Diagrams and Physical Diagrams. A

number of these approaches have been adopted to produce the UML model and are detailed further in the Chapter.

UML Class Diagram

Class Diagrams model class structure and contents using design elements such as classes, packages and objects. They are also used to display relationships that include containment, inheritance and associations.

Figure 4.3 – Simulation Class Diagram

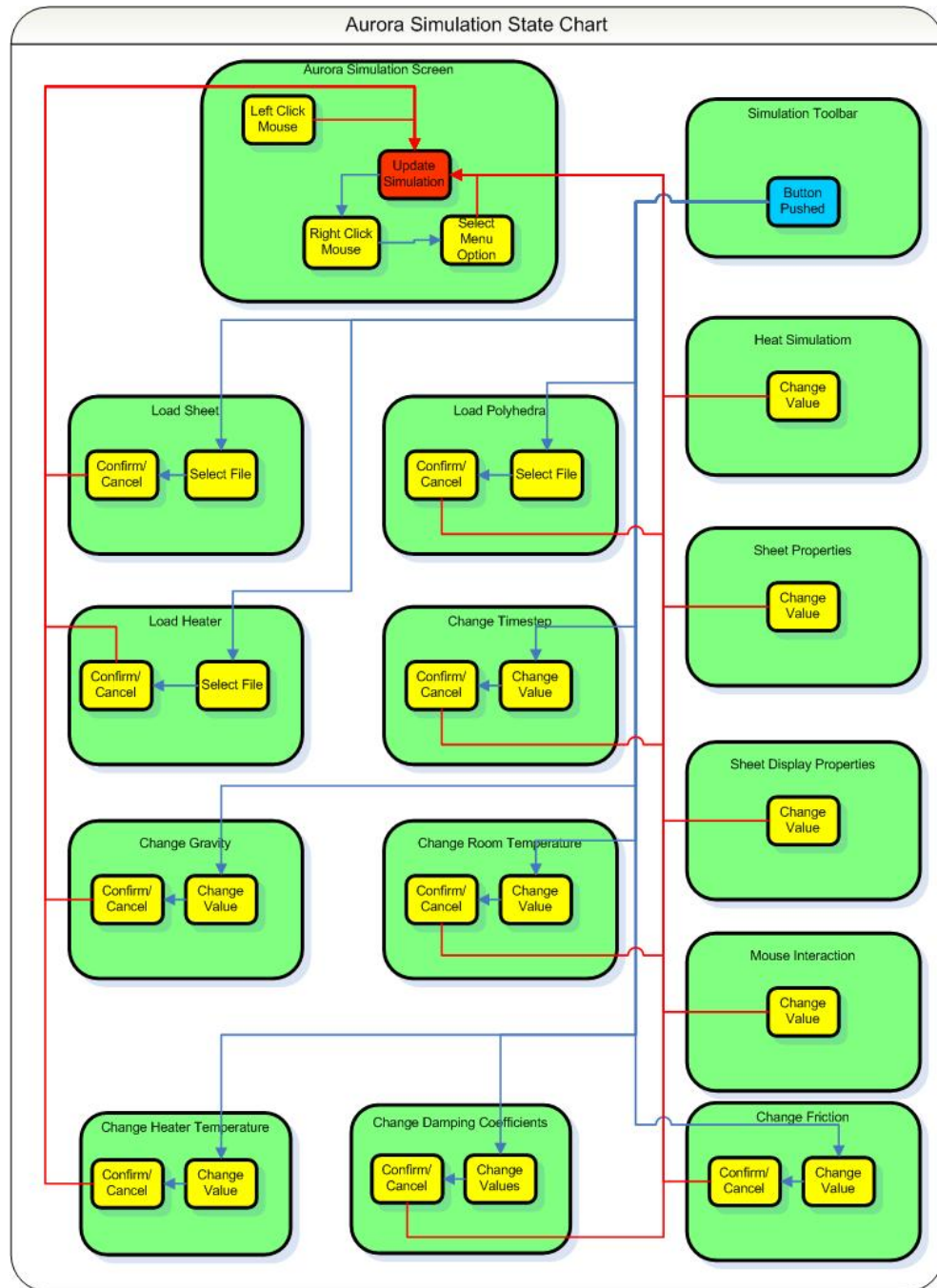


Interface System State Chart

State charts form an important element of the design process, which together with activity diagrams (see figures 4.5a, 4.5b and 4.5c) form the sequence of states that an object of an interaction goes through during its life in response to received stimuli, together with its responses and actions.

The state chart described in figure 4.4 details the user interface processes during the simulation run-time to adapt the simulation parameters and update the corresponding simulation behaviour (and view) to reflect these changes.

Figure 4.4 – Aurora Interface State Chart



UML Activity Diagrams

Activity diagrams act as special state diagrams where most of the action states and transitions are triggered by completion of the actions of the source states. These diagrams are triggered by the completion of the actions in the source states.

The activity diagrams for the Ordinary Differential Equation (ODE) calculation engines are illustrated in figures 4.5a, 4.5b and 4.5c that together form the core of the Aurora Simulation System.

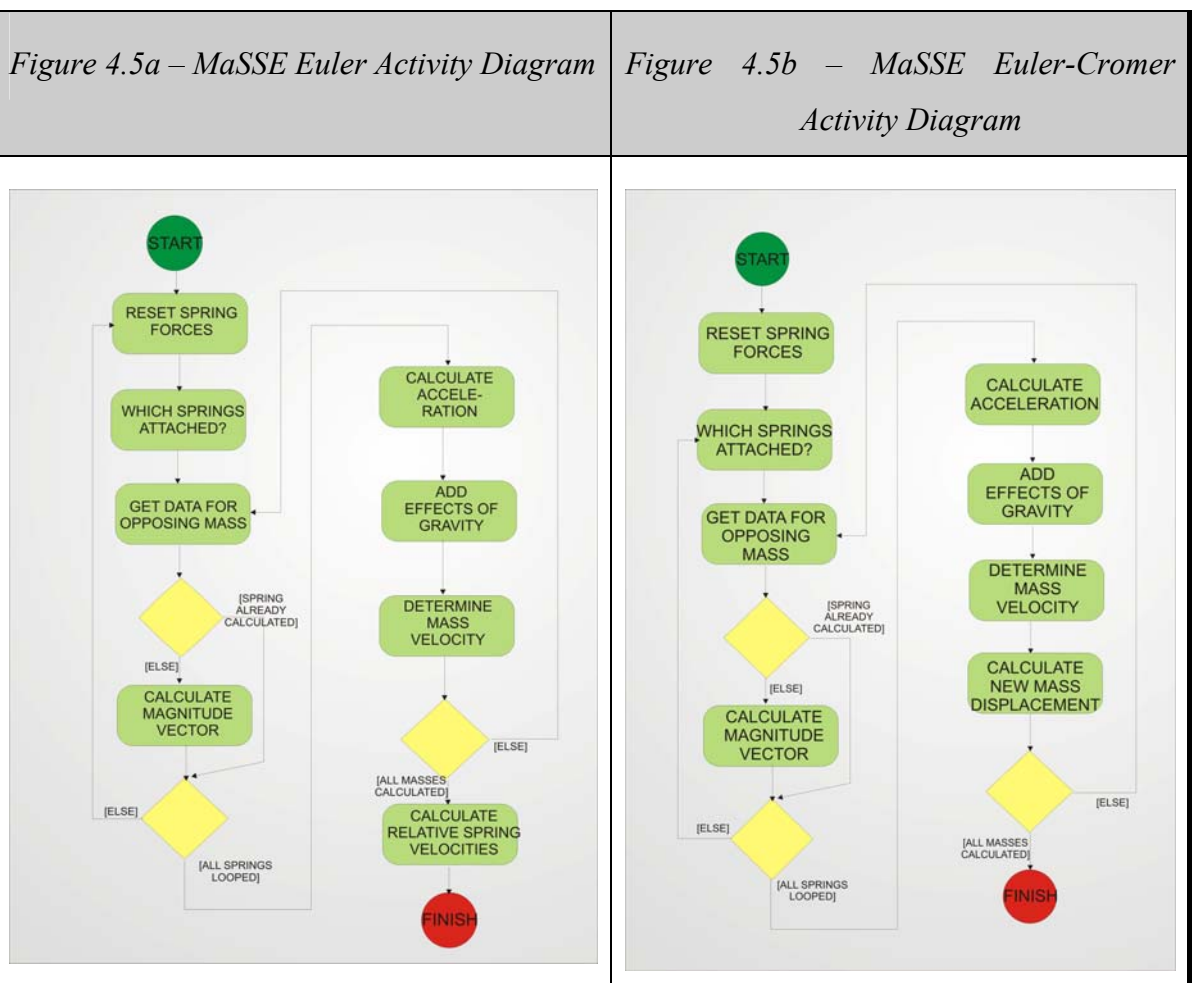
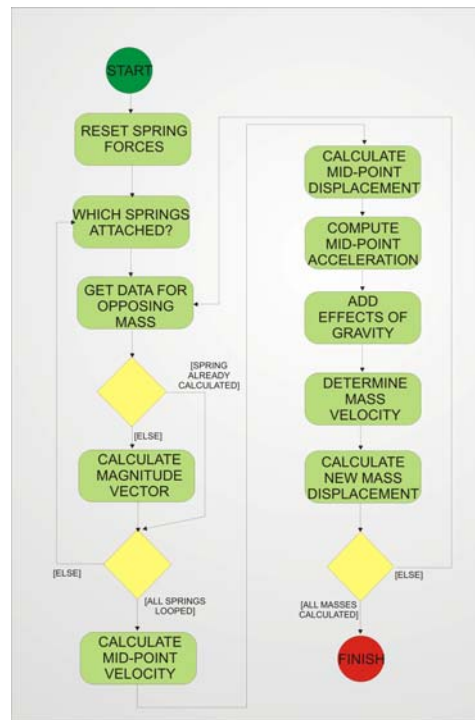


Figure 4.5c – MaSSE Euler-Richardson Activity Diagram



4.3 The System Software

To evaluate the methodology and results of this research, it was necessary to build an application suite on a PC platform, to test the concept of three-dimensional mass-spring structures. A PC platform was used because of its predominance within industry and its accessibility within both industry and academia.

The primary application is the sheet dynamics system (entitled ‘Aurora’) that forms the basis of this work. To cater for the complexity of creating the sheet structures and placing them in a configuration that Aurora can recognise, an automated sheet-creation program referred to as ‘Lightning’ was developed. The purpose of the remaining sections of this chapter is to outline the overall configuration of the system, specifically detailing how the Aurora application fulfils the requirements. Both programs were written in Microsoft Visual C++ as it provided the most comprehensive and mature development environment on the PC platform.

4.3.1 The Aurora System

Having based Aurora’s construction upon an object-oriented approach, the arrangement has been proven to facilitate a modular program structure. Although the initial overheads of such an approach can be significant, the long-term benefits include:

- Minimised program complexity.
- Enhanced system extendibility.
- Enforcement of structured design techniques.
- A logical mapping of real world objects, thus serving to create a natural appreciation of how the system was derived.

4.3.2 Class Structure

There are three principal structures that Aurora has been designed to accept as part of its simulations, these being a Sheet (referred to by the class *CSheet*), one or more Polyhedra (known as *CPolyhedra*) and a Polyhedra Heater (accessed through *CPolyHeater*). Each scene must, as a minimum, comprise of Sheet and Polyhedron structures that are inserted into the environment. Each of the three structures will now be examined along with their corresponding components which themselves form further classes.

Sheet

The *CSheet* class forms the basis of the three-dimensional sheet structure that is to be simulated. Like the actual (viewed) sheet, it is composed entirely of a series of interlinked Masses and Springs. Therefore, the *CSheet* class acts as a high-level structure of masses and springs in order to maintain an Object-Oriented format.

Mass

The class *CMass* is the primary component of the sheet structures, which is used to drive the behaviour of the sheet in relation to any forces. In its simplest form, a *CMass* acts as a reference point within the sheet. The ‘physical mass’ component of *CMass* incorporates an inertia value that is used to derive the mechanical oscillation of the sheet system.

Spring

To allow the *CMass* objects to interact with each other within the *CSheet*, a form of interconnect must exist between them. This binding link between masses provides overall cohesion and prevents each mass from independently floating off and it is this interaction that differentiates solid object model simulation from a conventional particle-based simulation. To fulfil this role the *CSpring* class is used to provide a spring force into the mechanical system, ultimately allowing changes in the distances between a given *CMass* and the *CMass*’s adjacent to it.

Polyhedra

An important direction of this study was the evaluation of the sheet as it interacted with solid objects in the environment. To aid this, the *CPolyHedra* class was devised to simulate one or more Polyhedral objects with which the sheet collides as it falls towards the surface of the simulated floor. Each Polyhedron consists solely of a number of faces that combine to form its shape. This is achieved through the use of pointers to each individual face object, providing access in a logical and structured approach. An additional parameter exists within the class to determine whether the polyhedron represents the floor of the ‘virtual room’ (for collision detection purposes), in which case it is not to be displayed but is dealt with elsewhere within the system.

Polyhedron Face

The face of a polyhedron object, referred to as *CPolyFace*, consists purely of vertex points that form the edge of the face. As such, the class contains no further data and acts purely as a join between the polyhedron object and the vertex class.

Polyhedra Vertex

The vertex class, or *CPolyVertex*, forms a position in three-dimensional space and represents the lowest level of the polyhedron hierarchy. These points are then combined to represent the edge of each polyhedron face.

Polyhedra Heater

One of the requirements of the system was the representation of the effects of thermal softening upon the sheet, which in turn led to the development of a virtual thermal source. One approach to this feature is the implementation of a heater that is based upon the same structure as the Polyhedra structures used previously. Thus, the *CPolyHeater* class was devised to simulate a polyhedron-based heater object that emits virtual heat from one or more of its surfaces. Each heater consists of a number of faces which when amalgamated form the overall heater shape. Again, this is achieved through the use of pointers to each heater face.

Polyhedron Heater Face

The face of a polyhedron object, referred to, as *CPolyHeatFace*, is not just comprised of the vertex points that form the edge of the face, but also (if it is specified as a ‘hot’ face) the parameters required to represent virtual heat (resulting in sheet thermal softening) within the environment.

Polyhedron Heater Vertex

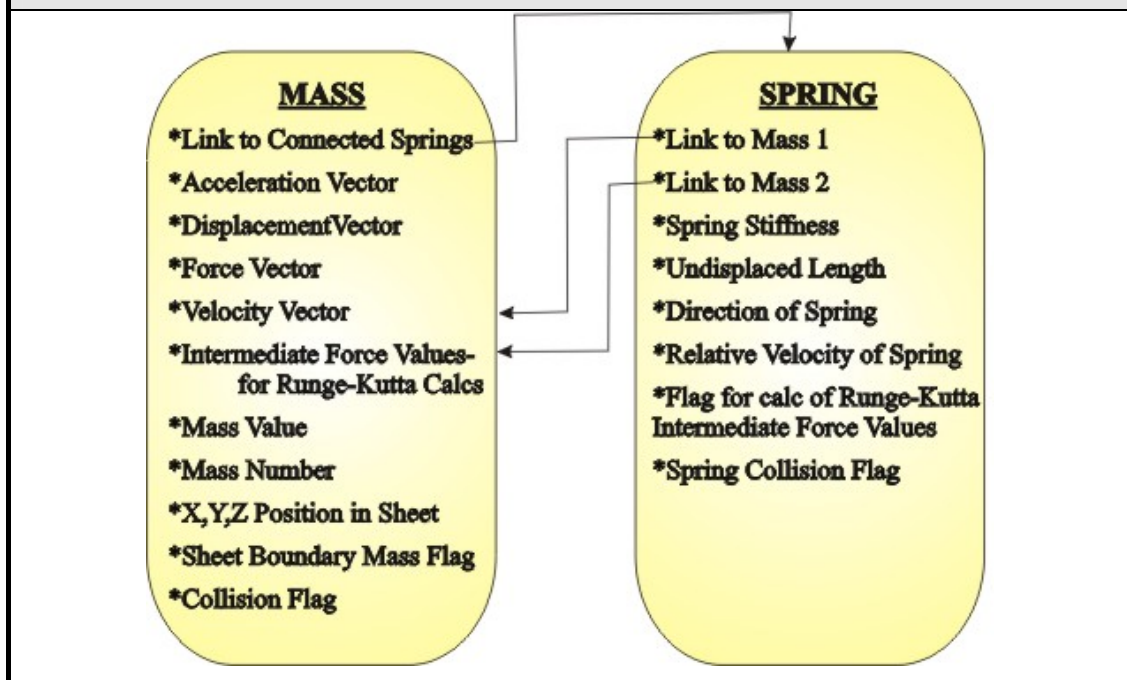
The vertex class, or *CPolyHeatVertex*, forms a position in three-dimensional space and represents the lowest level of the polyhedron heater hierarchy. These points are then combined to represent the edge of each polyhedral heater face.

4.3.3 Sheet Attributes

Each sheet is composed of mass structures that are interconnected by spring entities (see figure 4.6). Each *Mass* class first contains links to every spring to which it is connected, followed by data depicting the behaviour within free-space (i.e. acceleration, displacement, current velocity and any applied forces) as well as the specific weight mass value of the *Mass*. For the purposes of fourth-order calculations in predicting a *Mass*’s new displacement and velocity, using the Runge-Kutta approach, four intermediate results of the applied forces are stored. This topic is covered in more detail in Chapter 3. The final data includes details of the specific location of the *Mass* within the *Sheet* structure, its unique identifying number, whether it is positioned upon the *Sheet* edge, and an indication of whether the particle is in contact with any other object.

Each *Spring* initially contains links to the two *Masses* that it connects, as well as values for its undisplaced length (i.e. resting length) and spring stiffness. Other spring data includes the initial direction of the spring, its relative velocity and flag values to indicate whether the spring is colliding with any objects and (for the Runge-Kutta calculation engine) whether the intermediate force values have been calculated.

Figure 4.6 – Attributes of Mass and Spring Classes



4.3.4 Sheet Relationships

All masses and springs are stored in list structures that allow fast and immediate access to all *Mass* and *Spring* objects.

As discussed in section 4.3.2, each *Spring* contains a direct link to its associated masses, thereby providing a direct avenue to the data required to model the spring's dynamic behaviour. The benefit of this structure is that it allows the system to determine quickly the spring's length and position in free-space. This is done by identifying the displacement values of the masses, comparing the velocities of the spring's interconnecting masses and verifying whether the spring is contracting or expanding.

As a spring cannot exist without the two masses that mark its ends, the class structure ensures that each object closely resembles its underlying real world counterpart.

A reciprocal arrangement exists for each *Mass* that contains a list of each and every *Spring* that is linked to the object. Once more, this arrangement permits the system to perform calculations based upon the positions of its interconnected masses by dynamically accessing all of a mass's springs without the need to traverse sequentially each spring within the sheet's list of springs.

4.3.5 Lightning Project Structure

The purpose of the Lightning application is to create sheet files that are used with the Aurora system. There were a number of problems that prevented the sheet structures from being created individually by hand. These included:

1. The large number of Springs and Masses within a sheet. For example, a typical SLaMM sheet (see figure 4.8) with the dimensions of 11 masses (on the X axis) by 11 masses (on the Z axis) by 2 masses (on the Y axis) will contain 221 masses and 981 springs.
2. Each sheet structure contains additional '*system information*' that is required to perform the simulation. This includes identifying the type of sheet structure, the internal damping coefficient of the sheet, the positioning of corner and centre masses and specifying the sheet dimensions.
3. Allowing the production of complex sheet structures such as the Triangular TRiMM mesh structure (see figure 4.9).
4. In order for Aurora to fill each mesh correctly, each spring must be attached to each appropriate mass in a structured format, so that, for example the first spring to link to a mass will (provided that the mass is not located on the edge of the sheet) be in the X-direction, followed by all springs in the Z- direction and finally (provided the mass is not positioned in the top layer of the sheet) all Y (or inter-plane) springs. This format is covered in greater depth later within this section.
5. The software allows for the creation of dynamically stable sheet structures whereby all springs are initiated with spring lengths equal to the relative positions of their corresponding masses.
6. The ability to create multi-dimensional sheets with any number of (Y-axis) layers such as when creating three-dimensional cube structures (as opposed to laminar sheet structures).
7. Starting values for the sheet structure will be specified for all Mass and Spring objects within the structure. These initial parameters detail the initial sheet displacement and whether any Force, Acceleration or Velocity are applied to the whole sheet. Also included are values for spring stiffness, the mass for the Mass objects and the overall displacement volume of the sheet. This final value is then divided by the number of masses within each ordinate of the sheet structure to create the stable sheet structure.

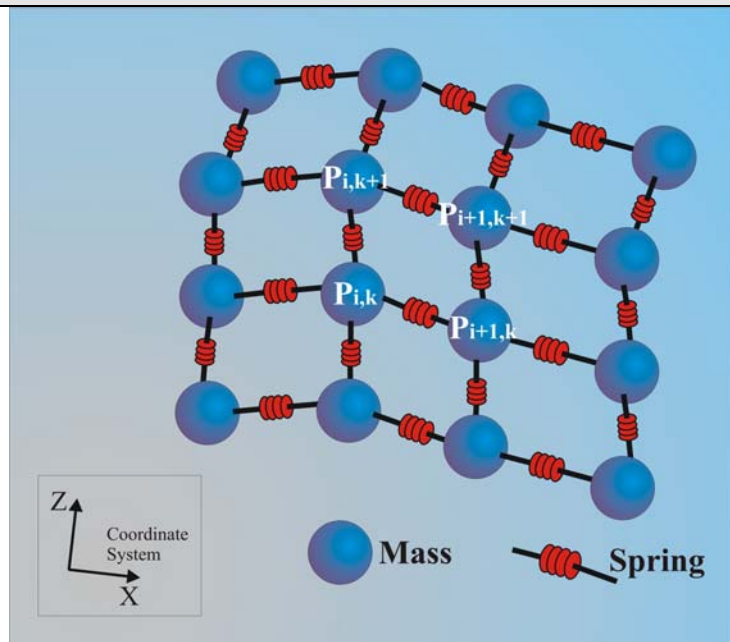
Limitations of the sheet creation facility are that

1. All structures are created in a uniform format and so creating a non-uniform sheet (with for example, one element (i.e. *Mass*) of a sheet being stretched in a particular direction) is not possible. To achieve this effect, the sheet data file must be amended using a text editor with the Mass's force, acceleration, displacement or velocity setting being amended as required.
2. It is not always possible to create a 100% stable sheet (due to limitations with specifying spring length in a text data file) so each spring may either contract or expand to its equilibrium (or undisplaced) spring length.
3. A generated sheet may only be horizontally placed within the environment, therefore, preventing the system from creating sheets that fall edge-on towards the surface.

The sequencing of each mass and spring within the sheet file is based upon a pre-determined order that is used by the simulation software to fill the mesh of masses with the virtual covering. This layout varies depending upon the adopted sheet structure with the SaMM, SLaMM and TriMM each comprising its own format.

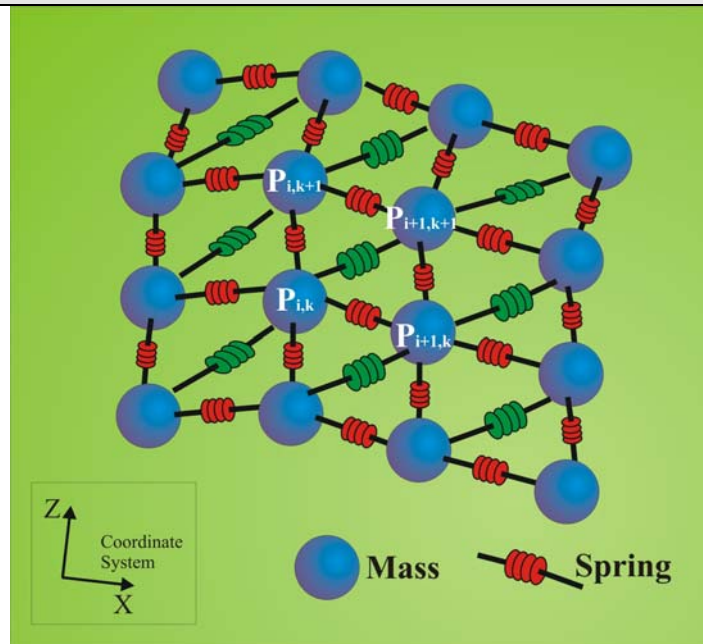
The basic structure of each layout is the traversal of every mass prior to the insertion of all the spring connections within the sheet.

Figure 4.7 – Mesh of SaMM Particles



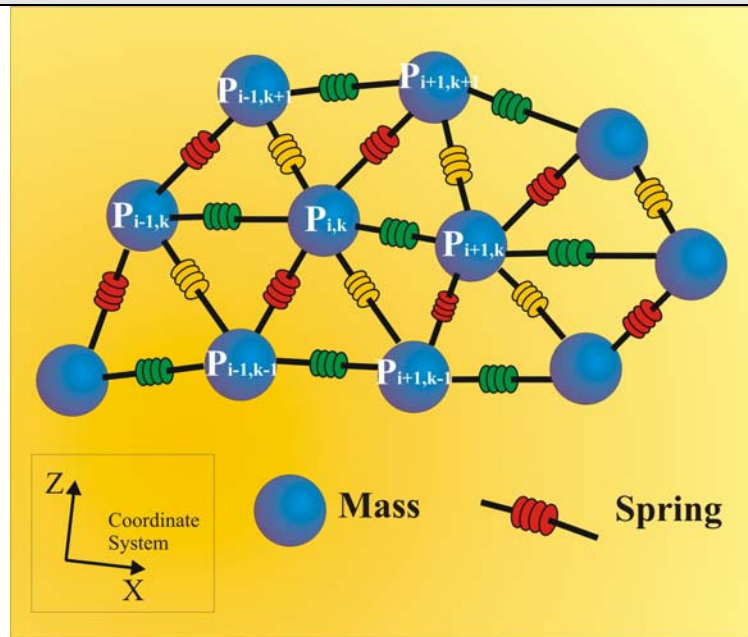
The SaMM system (see figure 4.7 and figure 4.11) determines the spring position of the current mass ($P_{i,j}$) and first checks that it is not positioned on the border of the sheet. If this is the case, the structure first appends a spring to the next logical mass in the sheet (i.e. the neighbouring mass in the X-direction, $P_{i+1,j}$). The next spring is used to link the current mass with the corresponding mass in the Z- (or depth) direction (i.e. $P_{i,j+1}$). Provided that the current mass is not in the top layer of the sheet, then springs are appended that link with the 4 bordering masses in the current spring's upper layer.

Figure 4.8 – Mesh of SLaMM Particles



The SLaMM system (see figure 4.8 and figure 4.12) works in an identical manner to the SaMM structure but with the added complexity of a cross-member spring (indicated by the green spring) that is used to link the current spring ($P_{i,j}$) with its forward-diagonal spring ($P_{i+1,j+1}$) before inserting the spring's depth, or Z-dimension spring ($P_{i,j+1}$).

Figure 4.9 – Mesh of TriMM Particles



The Triangular Mesh Matrix (TriMM) structure (see figure 4.9 and figure 4.13) is based upon a triangle format whereby the primary mass ($P_{i,j}$) first calculates the location of the next X-dimension spring ($P_{i+1,j}$) before determining the forward diagonal spring ($P_{i+1,j+1}$) and the backward diagonal spring ($P_{i-1,j+1}$).

4.4 The Sheets

4.4.1 The Sheet Data File Structure

The Aurora simulation is only able to cater for one inserted sheet at any time. Sheets are stored in a PC/DOS text file format using a simple structure to allow users to create/modify the structures easily and quickly without the need for specific applications or bespoke file conversion programs to interpret structures in the environment.

Each sheet file is denoted by the .SHT extension and is divided into sections comprising of Masses, Springs, Korner-Masses (that are used for collision detection purposes) and Sheet Parameter Data.

The formatting of the file components is described below.

Masses

Each mass is denoted by the 'M' indicator at the start (column 1) of the line, followed by 12 sets of floating point numerical values to denote the Mass's acceleration, displacement (or position), applied-force and velocity in three-dimensional space. The boundary conditions for the displacement values within the environment are as follows:

<i>Table 4.1 – Environment Dimensions</i>		
Dimension	Minimum Value	Maximum Value
X-Axis (10^{-2} Metres):	-6.5	6.5
Y-Axis (10^{-2} Metres):	-2.81 (level of floor)	3.0
Z-Axis (10^{-2} Metres):	-2.99 (position of back wall)	3.33

Additional data includes the mass value for the Mass, a sheet-level identifier and a sheet-border flag to indicate whether the Mass lies on the corner of the sheet.

Springs

The ‘S’ flag in column 1 on a new line denotes each spring object. The next parameter indicates the type/initial direction of the springs (for example in a SLaMM structure sheet, whether the spring is X-directional, Z-directional or cross-member) followed by data specifying the stiffness value for the spring and its undisplaced length.

Korner-Masses

The Korner-Mass flags are used to highlight the bottom layer sheet masses that form the four corners of the sheet, with a fifth value representing the closest positioned mass to the centre of the bottom sheet layer. These values are used for the purposes of collision detection.

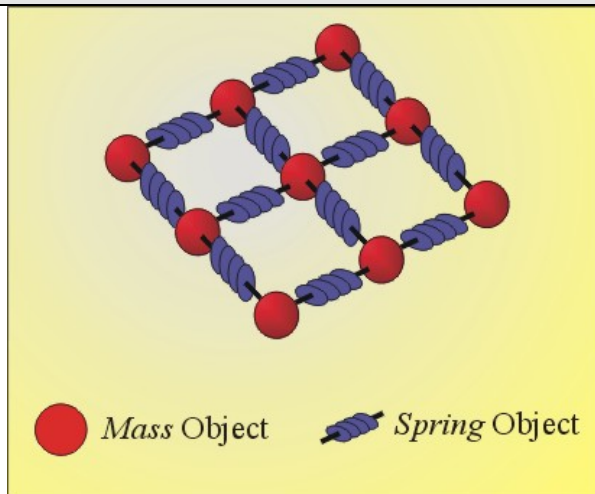
Sheet File Data Parameters

The end of file marker (denoted by a ‘*’ in the first column of a new line) must be followed by values for the Sheet Dimensions (preceded by the D flag), and then by a Sheet Damping Coefficient indicator and the type of Sheet Type ID.

4.4.2 Sheet Matrix Structures

Figure 4.10 represents a typical simple two-dimensional mass-spring implementation consisting of masses and springs in a two-dimensional layer (Gudukbay et al., 1997; Provot, 1997).

Figure 4.10 – Mass-Spring System Layout



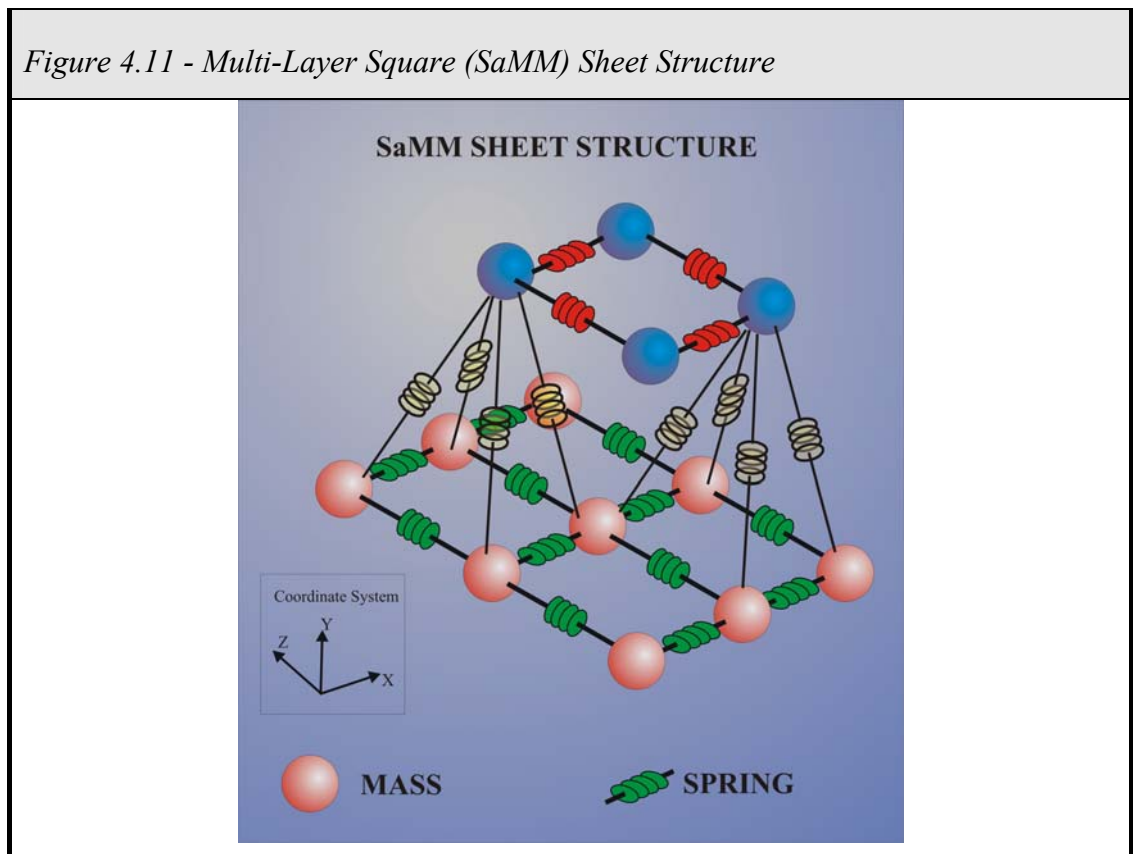
Like most elementary models, its basic structure gives the mesh a fast, efficient form producing acceptable results (demonstrating spring mass-spring interaction and elasticity) within a short run-time. However, since its two-dimensional nature (i.e. its lack of depth perception) results in a failure to account for bending moment, it was deemed an impractical choice for further investigation.

Bourguignon and Cani (2000) conducted a thorough investigation of implementing their derivative of mass-spring systems for use in anisotropic elastic materials, i.e. organic based bodies that maintain a constant or quasi-constant volume through the use of complex three-dimensional mesh structures. This was achieved by using cross-mesh axes that are used to support loads across the object, thus preventing the load from being carried along the mesh edges, thus helping to retain a constant volume for the object.

In total, three sheet structures have been devised and tested for the proposed system, and these are examined below.

SaMM

The square model structure, known as SaMM (see figure 4.11) is based upon a standard square-matrix sheet structure (see figure 5) with the addition of a third (depth) dimension, provided through the addition of a second (or multiple) layer of square matrices.

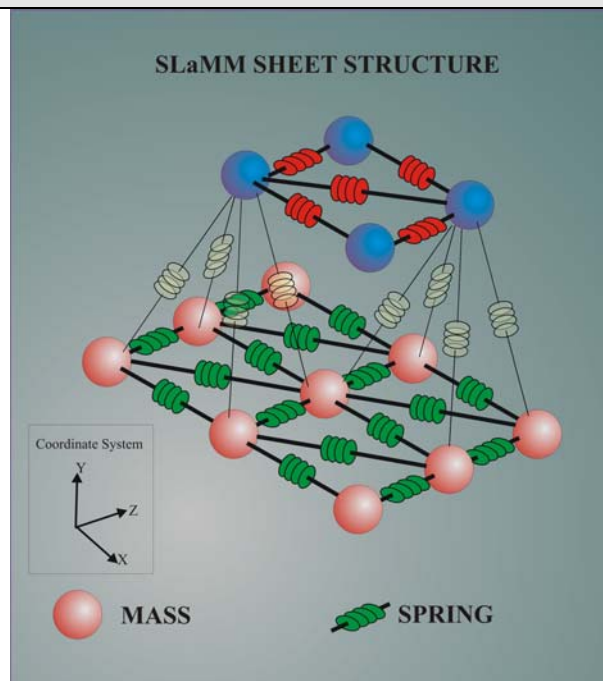


Each mass placed in higher layers of the sheet is positioned above the centre of the square formed by the masses in the layer below. Each mass in the *upper* layer is then interconnected to the layer below through four springs, which link it together with the four masses that form the square in the *lower* level. The advantage of this approach is that it caters for bending moment in the sheet (through the use of the cross-layer springs) while maintaining a high degree of efficiency.

SlaMM

The square-lattice model structure, known as SLaMM (see figure 4.12) is similar to the SaMM layout (see figure 4.6) but with the addition of a *diagonal* cross-member spring that interconnects the near-side left mass with its relative far-side right mass.

Figure 4.12 – Multi-Layer Square Lattice (SLaMM) Structure

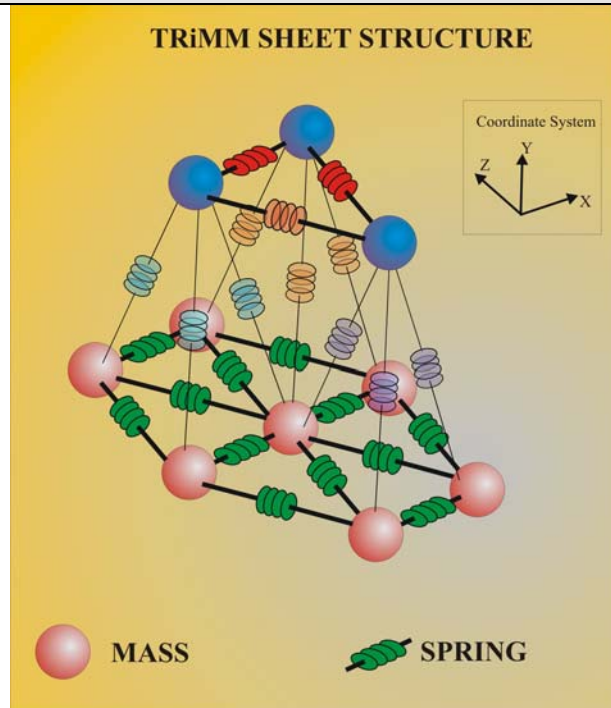


The role of this cross-member (or spring brace) is to prevent the square subsystems of the layer from forming a flexible mechanism, instead creating a rigid structure. Visually, the result of this additional spring is to prevent the square subsystems bending diagonally to form a rhombus. Triangulation of the mass-spring system has proven to be an important advancement in the design of the sheet structures, resulting in experimentation with the more unconventional TRiMM sheet structure that is described next.

TriMM

The triangular model known as TRiMM (see figure 4.13) evolved from extensive experimentation with the SLaMM model and the effectiveness of triangular structures within the sheet mesh.

Figure 4.13 – Multi-Layer Triangular (TRiMM) Sheet Structure



TRiMM represents a change from the two earlier structures through its use of a *triangular* base for the sheet layer. Successive layers are then positioned in an overlapping manner (similar to the SaMM and SLaMM structures). Again, each mass in the upper layer is connected (through a series of springs) to the lower layer via the three masses that form the triangle directly below which the mass is positioned.

4.5 Polyhedra

4.5.1 Data File Structure

The Aurora environment allows insertion of multiple Polyhedra objects, allowing a level of interaction between the sheet and its environment as it falls under the influence of gravity. This enables the sheet to deform upon collision with bodies in the environment, thus demonstrating the efficiency of the proposed model. As with Sheet objects Polyhedra are stored in a file format, facilitating easy user adaptation without the need for specific modelling applications or file conversion programs.

Each polyhedron file is denoted by the .POL extension and is divided into sections denoting a polyhedron, which are divided into faces that in turn are subdivided into vertices.

Each file is capable of storing multiple polyhedra. Each of these polyhedron may consist of an unlimited number of faces with each face capable of an unlimited number of vertices. The formatting of each file components is described below.

Vertices

Each vertex is denoted by a 'V' indicator at the start (column 1) of the line, followed by three float numerical values to denote its position in three-dimensional space. The boundary conditions for the environment are detailed in Table 4.1.

Faces

Following a series of vertices, the face identifier (denoted by the 'F' flag in column 1) shows the vertices that combine to form that Face. This will combine all of the previous Vertices since the previous Face indicator.

Polyhedra

Each Polyhedron, is identified by the 'P' indicator at the start of a new line. A Polyhedron must follow a Face indicator and is the sum of all Face indicators since the start of the file or (in the case of multiple polyhedra) the last Polyhedron indicator.

File

The end of file marker (denoted by a '*' in the first column of a new line) must follow a Polyhedron indicator, which in turn must follow a Face indicator.

4.6 Collision Detection

4.6.1 Overview

Standard collision detection algorithms result in an explosive growth of collision tests as the complexity (Zhong and Xu, 2009; Kim et al., 2009) and number of objects on the screen increase (Wilson et al., 1999; Liu, Ko & Chang, 1998; Vemuri, Cao & Chen, 1998). It is because of this that the WoW environment employs two approaches to provide what is believed to be the fastest and most efficient solution for the Aurora system.

Collision detection is a mandatory component of any animation system that assesses more than one moving object. Detection methods form a quadratic function of the number of moving parts in the environment together with their complexity (i.e. the number of faces per object).

A frequent explanation for the inactivity in assorted graphics-based disciplines, such as object deformation, is that one must first possess an accurate, cost-effective collision detection algorithm, which in reality has only come about in recent years. As collision detection methods have matured so has the capability to distort objects within synthetic environments resulting in a surge in deformation research activity.

Extensive work in collision detection over the past decade (Volino, 2005) has resulted in improved levels of precision by allowing avatars to wear multi-layer sets of garments (Zhong, 2009). Other areas of success have been: the successful adaption of collision detection algorithms with the Graphical Processing Unit (GPU) to enhance the simulation performance (Kim et al, 2009); helping to implement complex frictional contact between objects (Pabst et al, 2009), and more importantly, minimisation of the performance penalties incurred in using detection methods (Volino, 2005; Smith et al, 1995; Volino & Magnenat-Thalmann, 2000b). Ironically, the most successful results in collision detection have been achieved by groups whose primary goals were in other directions (i.e. object deformation and simulation), where a good collision detection algorithm was an important component towards achieving that aim. Based upon their own experiences in producing collision detection algorithms, Lin and Gottschalk (1998)

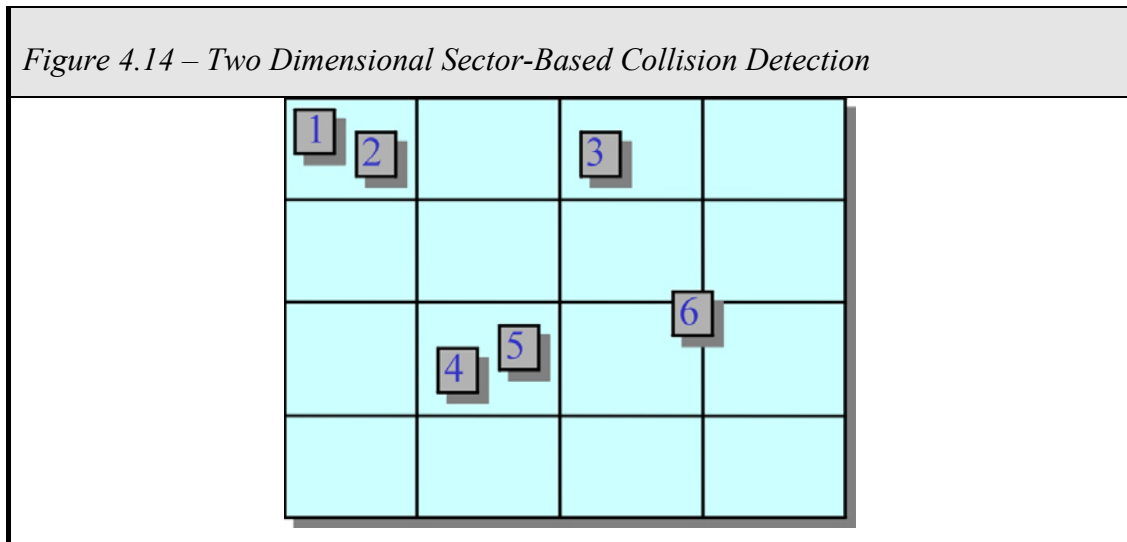
produced a thorough survey review of collision detection algorithms between geometric models. One of the more frequent examples of collision detection advancement as a by-product of other research is the complex modelling of cloth material as it interacts with external objects, gravity and itself (Teschner et al., 2005; Zhang and Yuen, 2002; Provot, 1997). Modelling clothes on virtual actors compounds this process further as its behaviour is determined by the material's interaction with the actor's skin. Due to its dynamic and deformable nature, particularly surrounding joints, any collision detection routine must first determine the new position of the skin as it flexes, even though it remains unseen by the viewer. Also to improve realism, the simulation must consider the type of material being worn, the cut of the clothing and even the weave of the fabric.

Since the 'early days' of simplified collision detection a number of complex algorithms that provide both fast and intricate levels of detection for novel applications have been proposed. These have included multi-core cloth based collision detection (Tang et al., 2009), as well as highly parallel architectures (Kim et al., 2009). Avril et al. (2009) conducted an evaluation of new trends and methods of enhancing collision detection performance with a significant focus being placed upon the use of GPU programming technologies and Parallel Processing. Ortega et al. (2004) have employed polar diagrams for their collision detection system. Their approach partitions the plane so that every scene object is the owner of a polar region as the locus of points with some common angle properties. However, this method is only suitable for collision detection work when applied during pre-processing when the solution is given by angle processing. Particle-based collision detection has always formed a challenging arena, with Senin et al (2003) utilising "sensor particles" that are distributed across a surface. This real-time method has proved an effective approach for simulating deformable surfaces.

Due to the level of research activities undertaken within the field of collision detection, a number of review papers have been published to help detail the distinguished approaches. One such example is the paper produced by Jiménez et al. (2001) that focused on collision detection approaches for three-dimensional environments.

4.6.2 Sector Based Testing

The first level of detection utilises game rules to imply that specific sprites[§] need not be collision tested against other sprites. In the current system these rules simply state that static sprites are not collision tested against other static sprites. As with many collision detection routines, the focus of interest is the sprites within the screen space.



The Sector method works by dividing the screen (or simulation view) into a three-dimensional grid of sectors. The algorithm then determines which sector each sprite is located in according to its screen position. A standard collision test is then made against any sprites that are within the same sector. Inefficiencies occur when a sprite lies on the boundary of two or more sectors, resulting in collision tests having to be performed between the sprite in question *and every other sprite positioned within any of the bordering sectors*. A simplified two-dimensional representation of sector based detection is provided in figure 4.14.

The virtual environment is divided up into a 25 (width) by 20 (high) by 15 (depth) matrix forming a total of 7500 segments. The system inserts each mass into the environment as a sprite and activates all of the sectors that contain any segment of polyhedra within its boundaries. Once a sprite enters an active sector, it undergoes

[§] objects on the screen that may be animating or moving

geometric testing against all plane surfaces within the room to ensure that it is not in contact with the surface.

4.6.3 Geometric Collision Testing

The second level of collision detection works by determining the equation of the plane for any surfaces in the simulation. The system then analyses the positions of all of the sheet control-points and (using the point-plane test) determines whether they lie above or below each of the simulation planes.

The test works whereby

$P = (x, y, z)$ = Point, N = normal to the collision plane = (A, B, C)

$$Ax + By + Cz + D = 0 \text{ or } N \cdot P + D = 0$$

$f(P) = 0$, if $f(P) > 0$, then the point P is above the plane.

If it is determined that the sprite (or more specifically, the *Mass*) has passed through a surface face, then prior to displaying its image, the positioning is adjusted to place the *Mass* directly onto the surface of the face. This new position is established by recalculating the position of the *Mass* based upon the angle of the surface.

4.7 Shading The Sheet

Two de facto approaches to shading are currently available for real-time graphics, those of Gouraud and Phong shading. These will now be examined.

4.7.1 Gouraud Shading

Gouraud shading is a method of simulating the differing effects of light and colour across the surface of an object. This has proven to be effective in achieving smooth lighting on low polygon surfaces without the heavy computational burden of calculating lighting for each individual pixel.

The principle of the technique is to calculate the surface normals at the vertices of Polygons in a three-dimensional model. The normals are then averaged for all the polygons that meet at each point. Lighting calculations (based upon the Lambertian diffuse lighting model) are then used to produce colour intensities at the vertices (Foley et al., 1990). The colour values are then interpolated along the edges of the polygons. The image is finally filled by the lines that are drawn across the image that interpolate between the previously calculated edge intensities.

Although significantly less processor intensive than Phong shading, Gouraud shading does not calculate all lighting effects as accurately (Angel, 1999).

4.7.2 Phong Shading

This is an interpolation of surface normals in rasterizing polygons to achieve better resolution of specula reflections. As Bui Tuong Phong in his published works combined the interpolation technique with his reflection model, “Phong Shading” is used to refer to the reflection model and/or a combination of the reflection model and the interpolation method.

The Phong illumination (or reflection) model is a local illumination model that can produce a degree of realism in three-dimensional objects by combining ‘diffuse’, ‘specular’ and ‘ambient lighting’ elements for each considered surface point.

Phong shading could be considered an improvement upon Gourald shading (Foley et al, 1990) as it is capable of providing a better approximation to a point-by-point application of an underlying reflection model by assuming a smoothly varying surface normal vector. However to take full advantage of its enhanced interpolation technique Phong interpolation should be used with the Phong reflection model, or any other specular-highlight based reflection approach.

4.7.3 Selection of the Shading Approach

Following experimentation using a simple Gourald and Phong Shading model, the additional computational complexity and load of utilising Phong shading failed to demonstrate significantly enhanced results within the visual demonstration environment. This is partly due to the unnecessary demand for specular reflection within the visual simulation, thus eradicating the need for the more advanced Phong Shading approach over its more efficient Gourald Shading.

Additional enhancements were gained through use of Gourald Shading using the embedded subroutines present within OpenGL and the GLuT toolkit thus minimising any computational overhead of adopting the technique.

4.8 The World on Window Virtual Environment

This section describes the construction of the World on Window (*WoW*) three-dimensional computer generated virtual environment. This setting forms the chassis upon which the sheet simulation experimentation work is based. The results of experimental work being discussed in Chapter 5.

The WoW virtual environment was created as a basis for real-time sheet simulation. The principal feature of this setting is the ability to view the animation from any desired three-dimensional perspective whilst retaining the ability to dynamically adapt simulation parameters. The system follows an intuitive design that eliminates the need for complex software interfaces or expensive, specialist hardware (Volino et al., 2005).

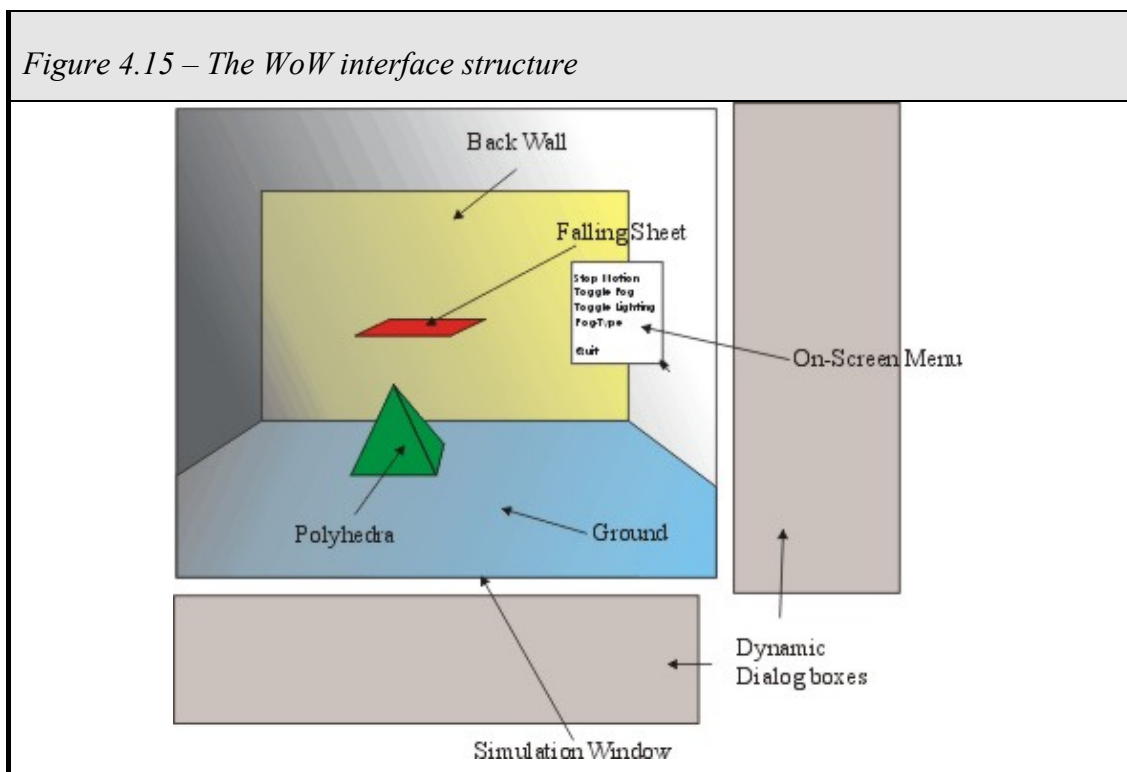
As discussed in Chapter 2, the adoption of a World on Window interface permits the simulation to operate on a wide range of PC platforms without the need for dedicated hardware, whilst retaining a highly effective degree of user interaction.

The system is comprised of two distinct components, the interface environment that is used to control the view and behaviour of the simulation (through use of the on-line dialog boxes and the simulation display window) and the kernel that links into the designated MaSSE engine and collision detection routines to perform the study of sheet behaviour.

The *WoW* environment is constructed using Visual C++ combined with the Microsoft Foundation Classes (MFC), the Open Graphics Library (release 1.1) which in turn uses the Graphics Library Utility Toolkit (GLuT) and the Graphics Library Utility Interface (GLUi). The system may operate on any 32 bit Microsoft Windows platform (e.g. Windows 7, Windows Vista, Windows XP) with a practical minimum specification (under XP) of a 300 Mhz Pentium II processor (or equivalent) and 256 MB of RAM. Any upgrade of this specification is likely to produce a significant enhancement in system performance or when using more recent operating systems.

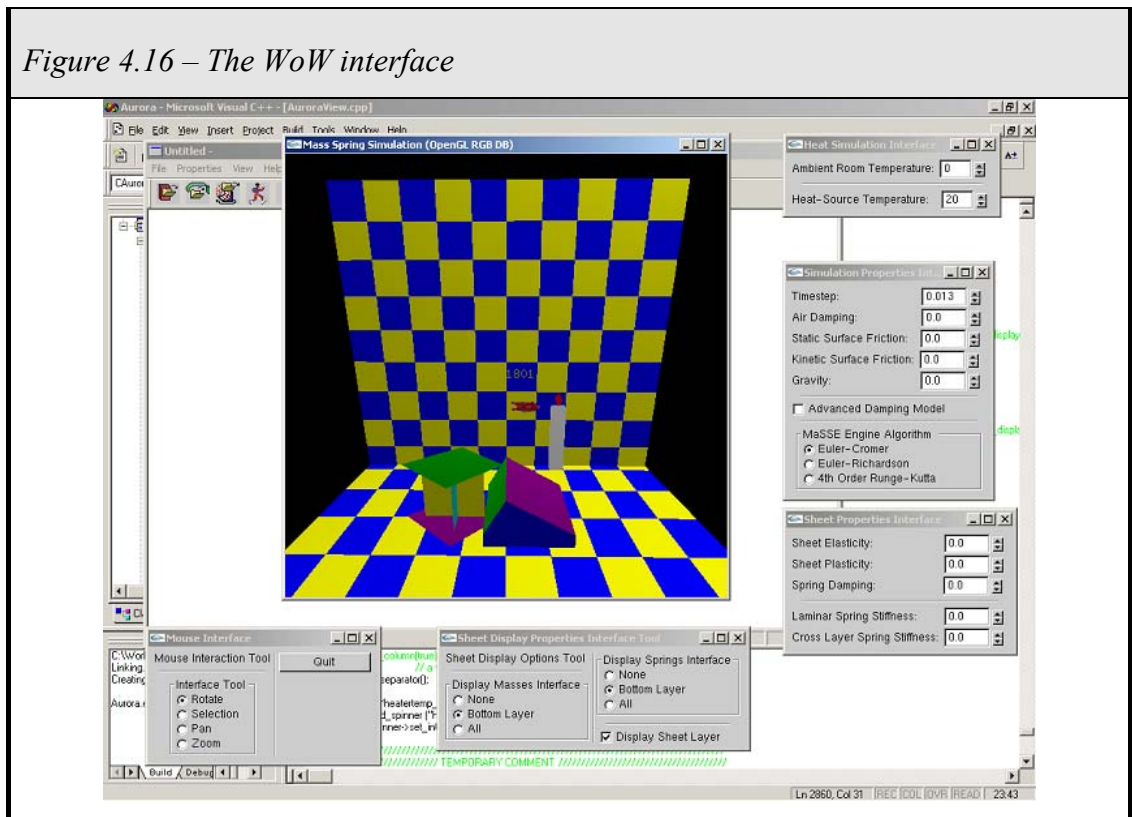
4.8.1 The World on Window Interface

The World on Window (WoW) structure is comprised of several distinctive parts. The primary interface is the simulation window (See Figure 4.15) which displays the simulation room environment. The room in its simplest form consists of a floor, back wall and is subject to gravity. During the simulation initialisation stage, the user selects and uploads a sheet, polyhedron and if required, an environment heater into the scene. Once running, the sheet will be viewed falling from its initial position, before colliding with the polyhedron. The capability exists to adjust a number of model parameters as the program continues to run. This is achieved either through the simulation window menu system (selected by clicking on the right mouse button on the simulation window) or through use of the dynamic dialog boxes.



The dynamic dialog boxes of the WoW interface (see Figure 4.16) enhance the capability of the software to be adapted to specified criteria. The interface has five dynamic dialog boxes that allow the simulation parameters to be adjusted in real-time.

Figure 4.16 – The WoW interface



These boxes consist of parameters for:

- The Mouse Interface
- This displays and controls the display mode of the mouse within the simulation window (i.e. Rotation, Pan, Zoom or None)
- Sheet Display Properties
- These control how much of the sheet is displayed (e.g. springs, masses or simply the sheet layer itself), whether the top or bottom layer of the sheet is displayed and whether the NURBS functionality is active.
- Internal Sheet Properties
- This displays and allows the user to adjust the internal damping coefficient of the sheet.

- Sheet Calculation Properties
- This menu determines the calculation engine for modelling the sheet behaviour (i.e. Analytical, 1st order Euler-Cromer, 2nd order Euler-Richardson or 4th order Runge-Kutta), and environmental parameters such as air-damping, time-step and friction.
- Virtual Heat Simulation Properties
- This controls the temperature of any virtual heat source (if present) and, if active, the ambient room temperature for the virtual heat facility.

4.8.2 The Viewing Perspective

The interface viewing environment is controlled by the use of a conventional two-dimensional PC-based two buttoned mouse. Rotation (or flight) around the environment is initiated by the user through depression of the left mouse button, combined with simultaneous movement of the mouse in the direction from which the user would like the new camera perspective. Alternative mouse-based view-angle functionality includes viewpoint Panning and the ability to Zoom in and out of the scene.

Use of the rotation facility is demonstrated in figure 4.17 where the movement of the mouse (with the left mouse button depressed) is highlighted by the blue arrow, resulting in the new viewing angle illustrated in the next frame.

Vertical mouse movements result in a change in altitude while a horizontal movement results in a variation of the lateral view of the scene.

The rotational facility is an integral component of the WoW environment, allowing the user to view the sheet deformation even when obscured by the polyhedral objects. To aid the viewing process the back wall and floor act as one-way mirrors thus providing a basis for the scene. This allows the user to view the sheet and Polyhedra when the viewpoint is behind or below the back wall or floor.

Figure 4.17 – Rotation about the scene

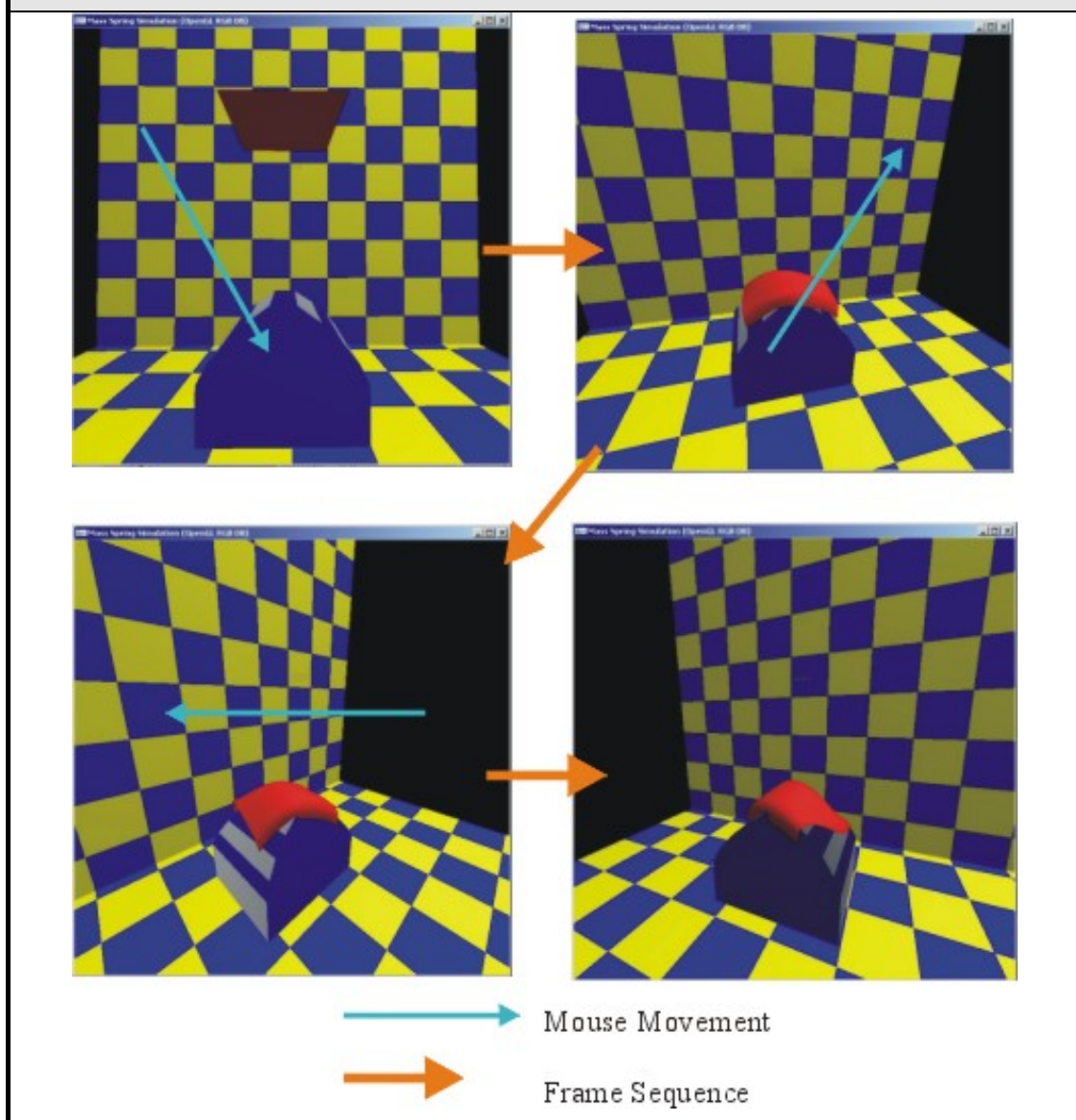
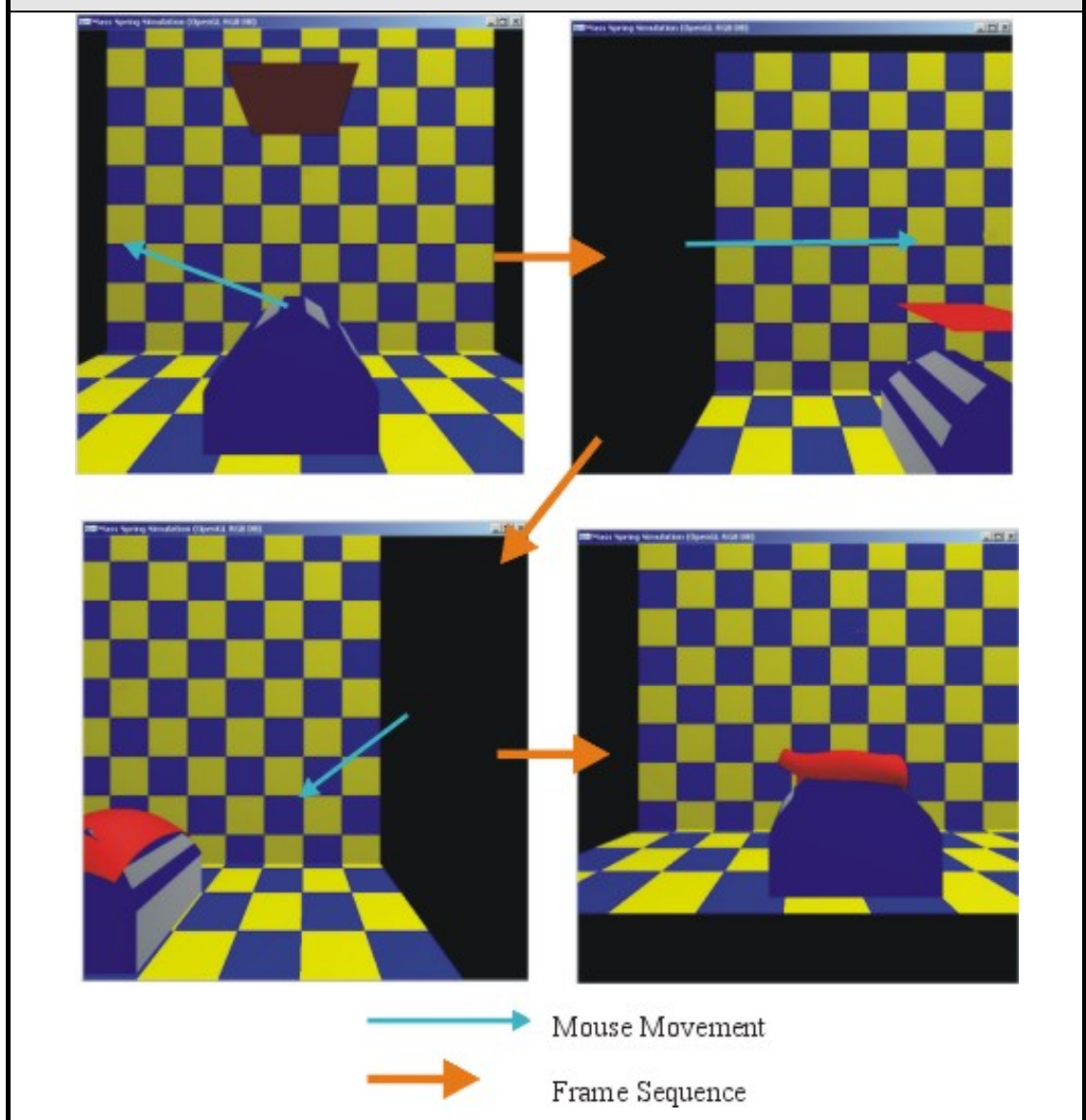


Figure 4.18 highlights the effect of use of the Pan facility. This ability is particularly useful when viewing the behaviour of a sheet, or partial sheet that is off-centre, or as a result of a collision, has become offset from the centre of the environment. On the following diagram, the movement of the depressed mouse is once more indicated by the blue arrows with the horizontal/vertical direction and length of the movement forming a corresponding change in the scene view.

Figure 4.18 - Exploiting the Pan facility



The capability of the environment to zoom in and out of specific sections of the sheet allows the user to focus upon specific areas of the sheet as it encounters a Polyhedron. The facility operates by the horizontal movement of the mouse, with a left movement resulting in the environment zooming out, to gain an overview of the complete environment. A movement towards the right of the screen operates the zoom-in facility that provides a close-up of any element of the simulation.

Figure 4.19 – Zooming in and out of the environment

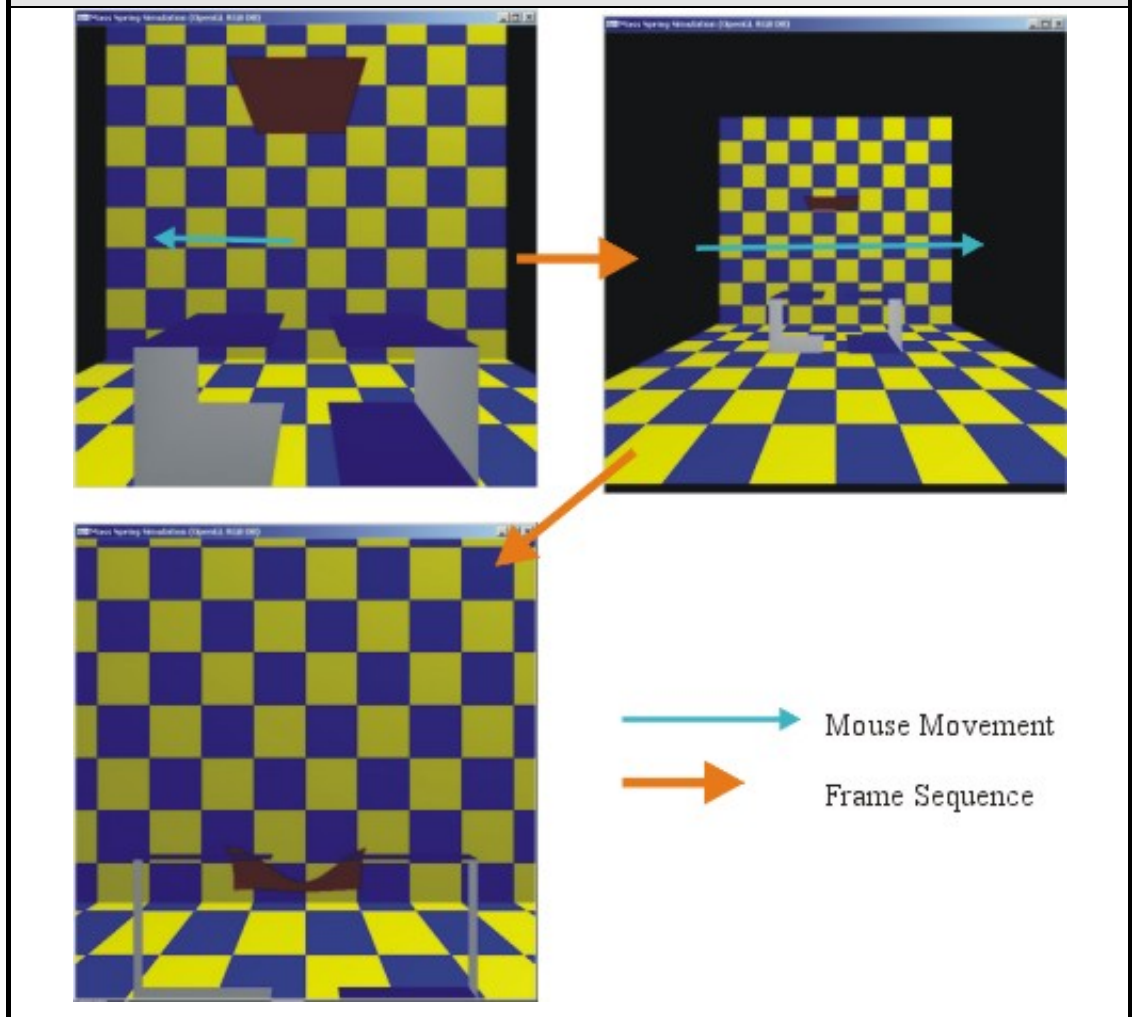
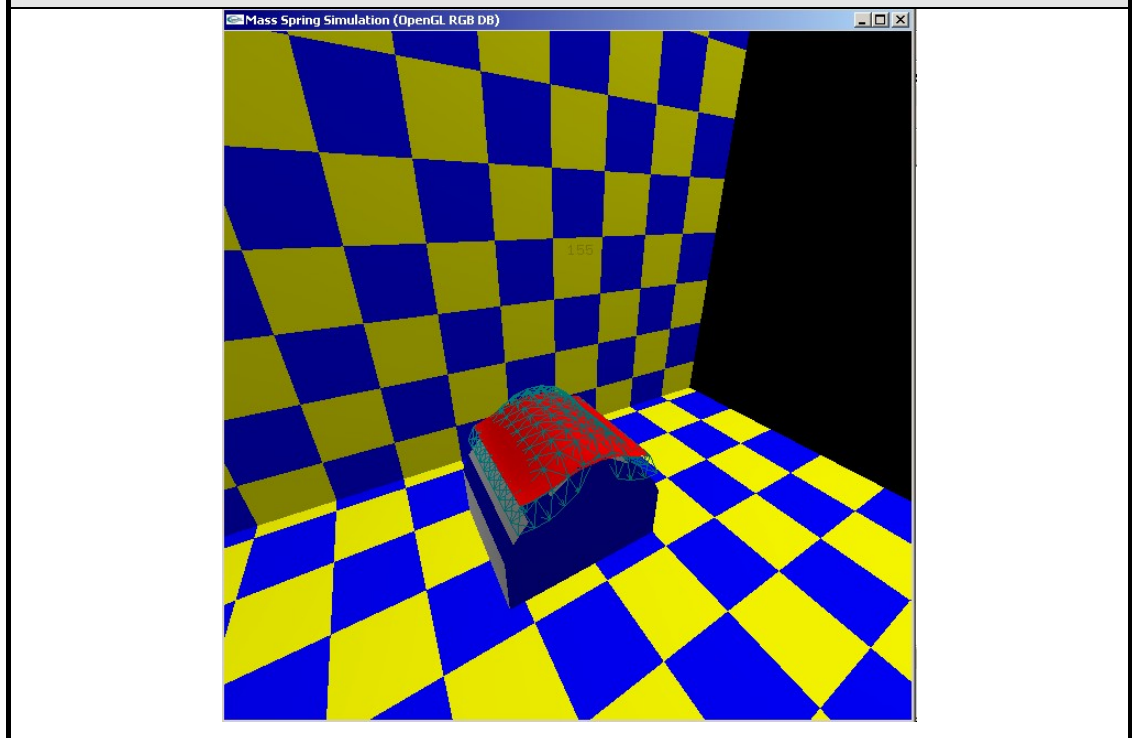


Figure 4.20 provides a typical view of the sheet simulation from the upper-left quadrant which is achieved through use of the rotation and zoom-out facility to gain an optimum viewpoint of the sheet as it deforms upon the polyhedron. In this example, the NURBS facility is active (resulting in the red layer) together with the option to view all of the sheet springs (coloured in green) for the SLaMM model.

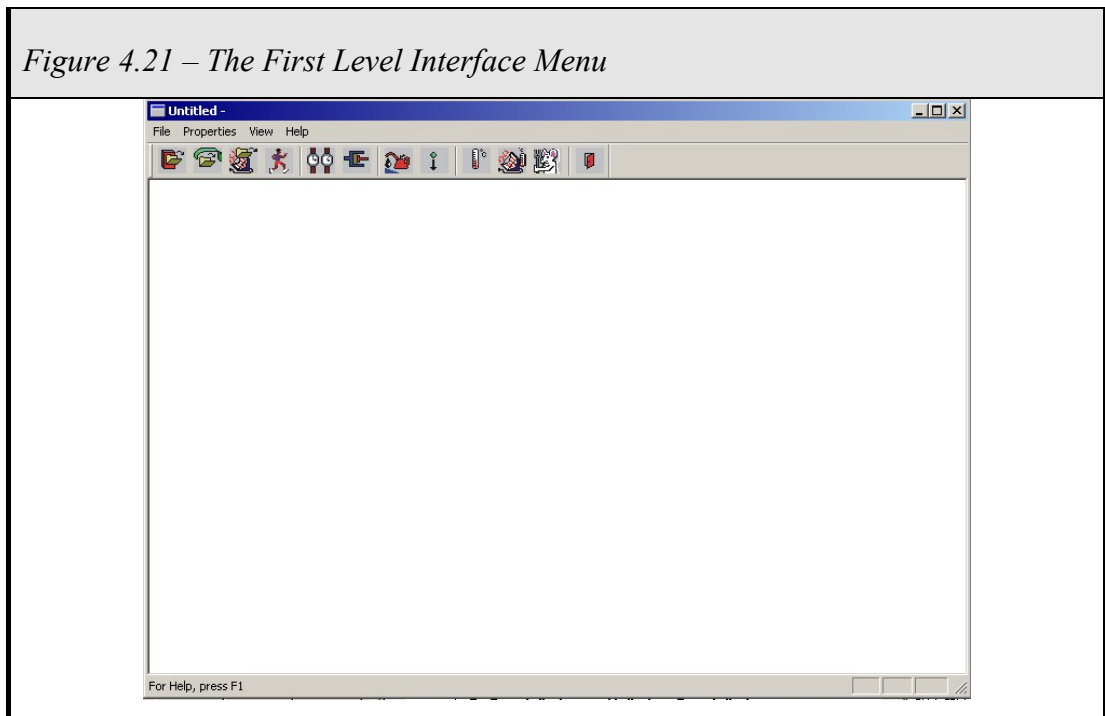
Figure 4.20 - Viewing a SlaMM sheet fall from an elevated left-side position



4.8.3 The Menus

Two levels of menu operate within the system. The first level is activated during the simulation initialisation stage, allowing the user to perform the minimum functionality required to perform the simulation. This consists of uploading a sheet, polyhedra and (if necessary) virtual heater files. Additional parameters include the command to run the simulation, adjust environmental parameters such as the simulation time-step, ambient room temperature and friction coefficient.

Figure 4.21 – The First Level Interface Menu



The second level of menus operates once the simulation has begun and is described in section 4.8 describing the World on Window Interface.

Setting the Scene

To demonstrate the ability of the sheet model to external stimuli, a decision was made to present the sheet model in a scenario (or scene) that would demonstrate the realism of the approach in a real world setting. A number of possibilities for this scene readily presented themselves, ranging from simple floor structures through to realistically rendered animations. Due to the computational expense of implementing fully rendered scenes, a compromise was devised based upon the uncomplicated yet elegant scene developed by David Yu (working for the Silicon Graphics Corporation) in his simple, 1992 spinning cube program, entitled Scube.

The scene contained the essential elements of a floor and rear-facing wall with no other identifiable features. Both the floor and wall are patterned in a simple square outline to act as subtle marker points and to distinguish the polyhedra and sheet structures from the surrounding background.

Lights, Camera, Action!

Simulation, like computer graphics works on the basis of modelling the system or scene internally within the computer, through the calculation of the object behaviour that is being modelled, before finally displaying the product of this work to the user. Prior to displaying the scene to the user, a number of processes are performed to enhance the look of the simulation to the user. These actions include rendering the scene through the addition of surface textures, lighting, and providing a viewpoint into the animation setting. The use of surface texturing and lighting are simple methods of providing a more realistic animation by instilling any surface with material properties (representing reflectivity and the visual smoothness of the surface), supplying lighting conditions that not only include placing the imaginary light source into any position within the room but also allowing for varying hues and intensity.

The final aspect of displaying a scene is the creation of a viewpoint. This is effectively the task of positioning a 'virtual camera' into the simulation that is utilised by the user to look into the scene. Like any camera, the user may govern the position, direction and magnification of the viewed picture through use of the Pan, Rotation and Zoom controls in the system.

4.9 Real World To Simulation Correlation

The purpose of this section is to examine the relationship between real-world environmental parameters and simulation variables to highlight any relationship that may exist. Table 4.2 examines this relationship for a group of typical simulation parameters.

<i>Table 4.2 – Real World Simulation Testing Parameters</i>		
Simulation Parameter	Simulation Value	Real World Environment
MaSSe Engine	2 nd Order (Euler-Richardson)	Not applicable
Time-step	0.20 (seconds)	Not applicable
Sheet Structure	Triangular Mesh Matrix (TRiMM)	Material structure (at a molecular level)
Sheet Size	11 wide, 2 deep, 2 long	Area of the sheet.
Distance Between Masses	0.125 (10^{-2} Metres)	Mesh density proportional to the force interaction area.
Restitution Coefficient	0.1 (10%)	Restitution coefficient and damping friction.
Kinetic Friction Coefficient	0.3	Dynamic friction.
Air Friction	0.97 (3%)	Air resistance
Gravity	9.81m/sec ²	0.981m/sec ²

Parameters for the simulation are detailed in Table 4.3 below.

<i>Table 4.3 - MaSSE Engine Testing Variables</i>	
VARIABLE	VALUE
Spring stiffness (planar springs)	185 N/M x 10 ⁻³
Spring stiffness (inter-layer springs)	185 N/M x 10 ⁻³
Mass value for each Mass	0.4 KG x 10 ⁻⁴
Sheet Size (location in room) – X dimension (min-max)	-1.4 – 1.1 (2.5 x 10 ⁻² Metres)
Sheet Size (location in room) – Y dimension (min-max)	0.25 – 0.35 (0.1 x 10 ⁻² Metres)
Sheet Size (location in room) – Z dimension (min-max)	-1.4 – 1.1 (2.5 x 10 ⁻² Metres)
Room Size – X dimension (min-max)	-6.5 – 6.5 (13 x 10 ⁻² Metres)
Room Size – Y dimension (min-max)	-2.81 (floor) – 3.0 (5.81 x 10 ⁻² Metres)
Room Size – Z dimension (min-max)	-2.99 (back wall) – 3.33 (6.32 x 10 ⁻² Metres)
Air damping coefficient	0.97 (3% coefficient)
Sheet damping coefficient	11.5% coefficient
MaSSE Engine	2 nd Order Integration
Time-step	0.015 Seconds

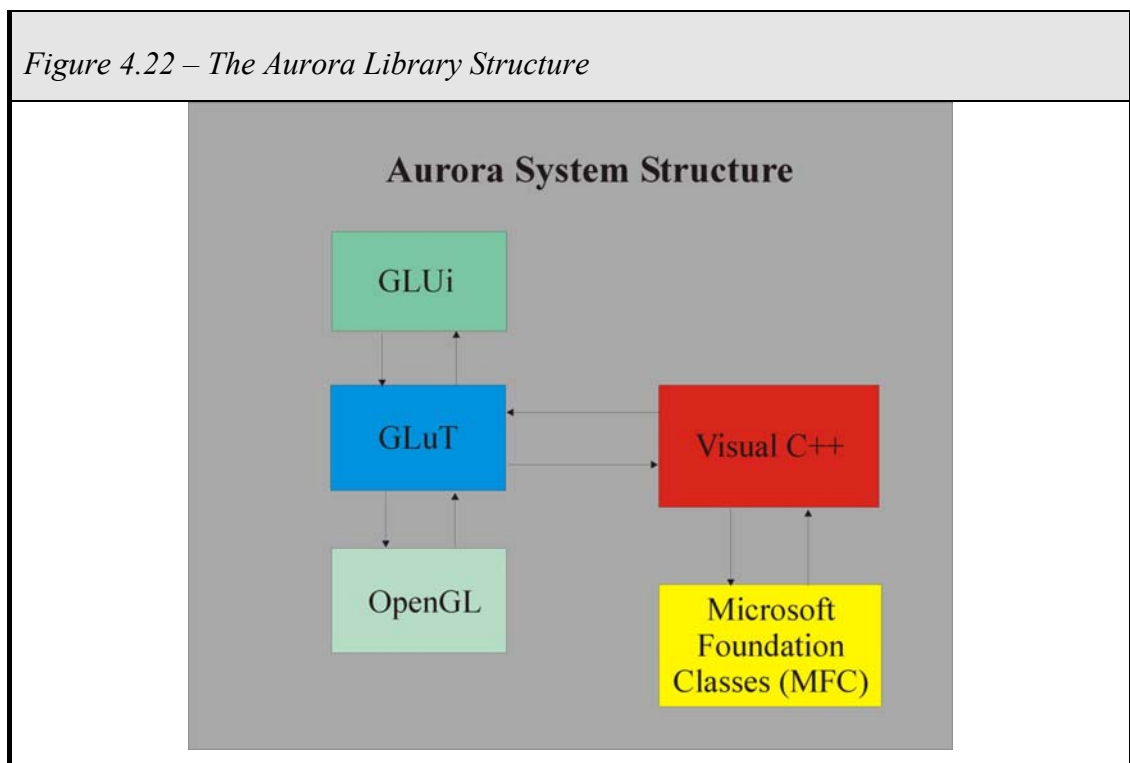
4.10 Aurora Application Library Structure

4.10.1 Overview

Aurora was designed using a number of software based libraries that interact with the core simulation software. The main purpose of these software routines is to:

1. Supply graphical routines to display the results of the simulation in the most aesthetically pleasing light and program well versed algorithms and routines into the system. This removes the need to recreate existing and well-documented software techniques.
2. Allow the operator to set-up initial conditions and parameters prior to initialising the simulation
3. Allow the operator to adjust the environmental and system parameters whilst the simulation is running.

Figure 4.22 – The Aurora Library Structure



4.10.2 Library Components

As previously described earlier in this chapter, Visual C++ forms the central hub of the Aurora system. It is used to bind together the system components, performing all mathematical calculations, calling library routines that are used to retrieve user parameters and utilise graphical routines to enhance the display of calculation results.

The Microsoft Foundation Classes (MFC) are closely integrated into Visual C++ and are regarded (by Microsoft) as the natural interface for its programming suite of applications on the Windows platform. As such, MFC was used to gather user parameters prior to the start of the simulation program, although all data gathered once the simulation has started its execution cycle is obtained using the OpenGL based GLUi interface library.

Using the Open Graphics Library (or OpenGL) formed a key project decision which subsequently dictated the fundamental structure of the system. Alternative graphics libraries were considered such as Microsoft's Direct X system although its reliance upon the Windows operating system and thus lack of portability across different platforms and operating systems proved to be a major shortfall. A description of the assessment procedure is covered in Appendix A.

The Graphics Library Utility Toolkit (GLuT) is used due to its effectiveness as a high level interface to the OpenGL libraries. Originally devised by Mark Kilgard (1994) whilst working for the Silicon Graphics Corporation (SGI), it is now regarded as 'the' de facto high-level toolkit for interfacing with OpenGL, providing a seamless interaction with the OpenGL sub-system.

The Graphics Library Utility Interface (GLUi) was integrated into the system due to the need to provide significant level of dynamic user interaction during the running of the simulation. This was not available within the standard GLuT or OpenGL libraries.

4.11 Thermal Softening Modulus

Heat transfer, the flow of energy from a high to low temperature region, occurs in almost every phase of scientific and engineering work. The implementation of virtual heat transfer within virtual environments, however, has been a dormant topic within the VR community. This is a basic problem given that one of the principal applications of synthetic environments is the simulation of hazardous environments, for the purposes of training and traditional forms of virtual walkthrough. One early example of this was conducted by Bukowski and Séquin (1997) who developed an interactive virtual environment based simulation of fire sweeping through a building, analysing the behaviour and toxicology of the smoke as the building and its contents are combusted in the fire. The CFAST system then allowed comparisons to be made in the behaviour and direction of the fire as doors are opened and closed, although the system was incapable of producing realistically accurate models of smoke behaviour as it swirled around the room and interacted with the room contents.

To date, industrial-based engineering simulations have dominated current research in simulated plastics and other materials as they are exposed to a heat source. Since traditional research in virtual environments has concentrated upon thermally static environments, thermodynamic simulation has remained firmly entrenched in mathematical, engineering and physical science based disciplines.

These areas will now be studied in further detail.

4.11.1 Animation of Virtual Heat

As outlined in Chapter One, one of the unique qualities of the environment presented is the ability to reproduce real-time thermal softening in the behaviour of the simulated materials. To accommodate this, the following schema has been devised, prior to its implantation into the environment.

A value is given to the sheet material that defines its thermal sensitivity (its susceptibility to plastically deform due to the effects of exposure to heat), this is in addition to a value

for the ambient temperature is also added. At present, assumptions are made that the room will be set at an initial constant and homogeneous temperature with no allowances made for additional effects such as convection or radiation (e.g. strong light). This differs from conventional methods of calculating heat that require complex CFD calculations that determine the flow of heat as it circulates around the room under the influence of convection currents.

The representation of thermal softening based deformation upon the sheet caused by the application of the virtual heat is achieved by weakening the stiffness of all the sheet springs. The levels by which the stiffness values are reduced are dependent upon the type of heat source being applied, the magnitude of heat and, if using a heat-source, the distance of the heater to the sheet.

Two forms of heat transfer are available within the Aurora environment. The first is a radiation based direct virtual heat source, represented by a virtual heater that is loaded into the room, and the second mimicking the room having been flooded with virtual heat energy. These mechanisms differ significantly in their method of reducing the strength of the sheet material.

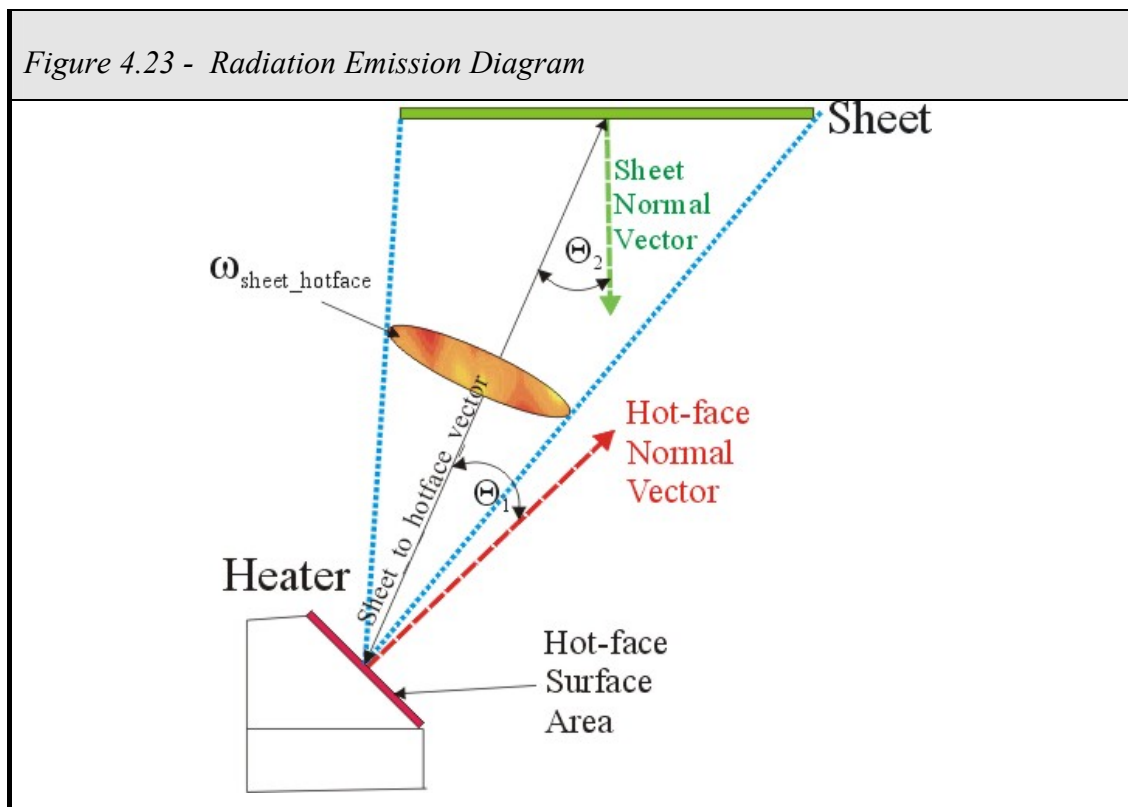
Oven Heat Representation

The oven thermal softening method operates by weakening all the springs throughout the sheet, regardless of their position within the sheet mesh and distance from the walls of the room. The oven facility may be activated or deactivated at any time during the operation of the simulation by selecting the oven heat activation switch on the 'Heat Simulation Interface' dialogue box. The user may then update the temperature within the virtual room to vary the amount of heat present, thus altering the amount of sheet deformation. The default room temperature has been set to twenty degrees Celsius as any thermo-sensitive material unstable at this temperature will be affected by temperature fluctuations. An increase in temperature will amplify the levels of plastic deformation and a reduction below room temperature will cause increased sheet solidity.

Heat-Source (Radiation) System

The thermal softening functionality of the application is a variation in modulus that is a function of temperature. The heat flux results in a change in temperature that is dependent upon the thermal conductivity and capacity of the material.

The method for predicting the magnitude of virtual heat being encountered by the sheet surface (and thus the degree of sheet thermal softening) uses the virtual heat-source mechanism as shown in figure 4.23.



Once a virtual heat source is activated and set to a specified temperature calculations are performed to compute the exact distance of the centre point for each spring to the virtual heat source. This gap (or Spring Centre Point - SCP) is calculated by first determining the interval between each of the two control points that enclose the spring and then enumerating the length mid-way between these points to the virtual heat source. The centre position of the virtual heat surface's hot-face(s) is then calculated together with a value for the heat-face's total surface area. Like polyhedron faces, the virtual heater may have up to eight edges per face.

The level of emissions being intercepted by the sheet from the hot-face is then calculated, based upon the angle of the sheet relative to the hot-face, the intensity of virtual radiation being emitted from the heater and the distance being travelled by the virtual radiation.

This is enumerated (see figure 4.20) as

$$Rate = Intensity \times (Area \times \cos \Theta_1) \times \omega_{Hotface} \quad (\text{Equation 4.1})$$

where

Rate = Rate of Radiation Intercepted by the sheet

Intensity = Radiation Intensity

Area = Area of Sheet

And

$$\cos \Theta_2 = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \frac{(A_x)(B_x) + (A_y)(B_y) + (A_z)(B_z)}{\sqrt{(A_x)^2 + (A_y)^2 + (A_z)^2} \times \sqrt{(B_x)^2 + (B_y)^2 + (B_z)^2}} \quad (\text{Equation 4.2})$$

And

$$\omega_{Hotface} = \frac{Area \times \cos \Theta_2}{(Distance)^2} \quad (\text{Equation 4.3})$$

where

A = Vector from the sheet centre to the normal vector of the hot (heater) face

B = Vector from sheet to the hot (heater) face

Distance = Distance from the hot (heater) face centre to the sheet centre

Once a value for sheet radiation has been achieved, a linear calculation is then performed on each spring, adjusting its stiffness, based upon the:

1. Current spring stiffness (K),
2. Distance of the Spring Centre Point to the heat source (d),
3. Coefficient of 'heat sensitivity' (capacity) of the sheet material (C),

4. Radiant energy generated by the heat source (Ht),
5. Ambient room temperature (At).

If the modulus of heat is described as E , material thermal capacity = C , spring stiffness = K , distance = d , Heater Temperature = Ht and Ambient Temperature = At then

$$E = K - \frac{(Ht - At)}{d} C \quad (\text{Equation 4.4})$$

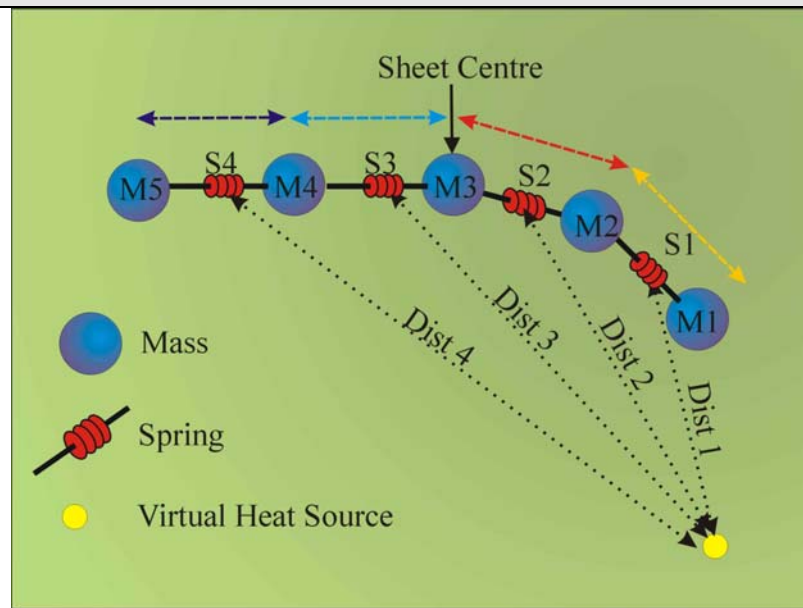
The material's Thermal Sensitivity Coefficient represents the susceptibility of the material to thermal softening. Its value determines the overall sensitivity of the material to the total radiation being encountered by the sheet before dividing the effect equally across the entire sheet.

A second parameter, known as the Thermal Conductivity Coefficient acts as part of the Pseudo-Conductive Heat System, providing each spring with a local heat-sensitivity autonomy based upon its positioning within the sheet and distance from the heat source (see figure 4.24). The system determines its position in relation to the two points and adjusts the level by which the spring's stiffness value is to be reduced depending upon these values and the coefficient value representing the material's thermal conductivity.

By altering the rate of spring (stiffness) weakening, the Thermal Conductivity value increases the level of deformation in sheet areas that are positioned closer to the heat source, reducing the levels of deformation on outlying areas. The system works by examining the position of every sheet spring relative to the sheet centre and the focal point of the heater face. As a result of the spring's position relative to these points, the outcome of the Spring-Weakening system is proportionally amended to take into consideration the results of the Thermal Conductivity algorithm, dependent on its distances between these two points and the actual Conductivity value.

Although the system caters for varying exposure to radiation from the heat source across the sheet surface, a further area of study is the adoption of virtual heat transfer within the sheet itself (i.e. conductivity based softening) without the use of thermal convection.

Figure 4.24 – Demonstration of the off_centre_spg_weakening coefficient



4.12 Summary

This chapter described the class hierarchy and structures of the Aurora sheet and Lightning sheet-generation applications before discussing the unique properties of the three-dimensional sheet structures that operate within the Aurora simulation. Having outlined the basic file structures of the sheets and polyhedra, the current state of the art in collision detection algorithms were examined, together with the sector based and geometric testing systems implemented in the sheet deformation simulation software. The chapter then analysed NURBS curves and its use to produce smooth sheet curve deformation before describing the structure of the World on Window (WoW) environment and its Human-Computer Interface. Finally provided was an overview of the overall Aurora system structure and how its various libraries interact with the C++ development software.

The implementation of thermal softening into virtual environments has proven to be a new and untapped field of investigation. Other than engineering-based thermodynamic simulations, most non-haptic based research has tended to focus upon producing aesthetic simulations of fire and the behaviour of heat convection currents as they manipulate their ‘virtual’ heat particles around a room. These virtual particples are used to visualise the flow of heat around a room under the influence of convection forces. The heat system presented here represents a novel approach to mimicking the effects of a virtual heat source within the environment, most notable by its use of virtual heat to produce an actual affect on items within the WoW environment through the act of deformation upon the sheet. Although limited in its mathematical accuracy, the system has proven to form a sound foundation upon which to develop virtual-heat into mass-spring-based systems.

5 Experimental Testing

5.1 Introduction

The work described in this chapter falls naturally into two distinct parts. The first examines the results of real world experiments that were conducted by studying the behaviour and deformation pattern of real plastic sheets as they fall onto polyhedral objects. Two aspects of their deformation are studied in this work, the first centres upon sheet bending at room temperatures while the second analyses the effects of thermal softening on the deformation process.

The second stage of experimental work is centred upon the calibration of the simulation software to match the behaviour of the real world experimental work.

This work provides the data required to allow the remainder of the chapter to draw comparisons between the behaviour of the virtual sheet to that of the real world lamina at both room and higher temperatures.

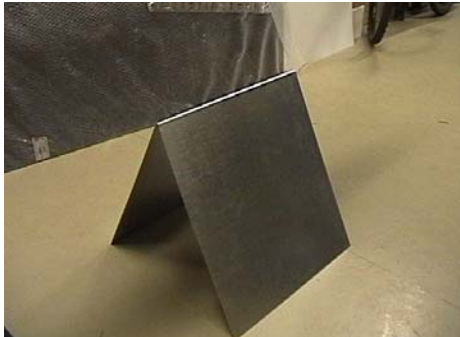

5.2 Real world Experimentation

5.2.1 The Polyhedron

Each polyhedron is constructed from 5mm sheet steel that is placed onto the test rig at room temperature.

Two polyhedral shapes act as collision objects during the experimental testing. The choices of shapes were based upon rudimentary geometric objects designed specifically to elicit a wide range of examples of sheet deformation behaviour.

These objects, which are shown in figures 5.1a to 5.1b respectively, comprise a triangular wedge and a partial cube.

<i>Figure 5.1a – Triangular Wedge</i>	<i>Figure 5.1b – Partial Cube</i>
	

5.2.2 The Sheets

In total, four sheets made from differing plastic materials were used during the initial experimental work. The purpose of this testing was to determine which form of material provided the closest representation of deformation to the virtual sheet model. The materials comprised of:

- 0.4 mm PVC
- 0.25 mm styrene
- 0.5 mm styrene
- Packaging/Industrial Grade PVC

These were filmed, at room temperature, falling onto the two different polyhedral structures; shown in figures 5.2 to 5.6.

5.2.3 The Room Temperature Experiments

Each sheet was filmed being dropped from a height of approximately one metre above the polyhedron, with results being presented in figures 5.2 through to 5.6. Figures 5.2 to 5.5 are used to visually demonstrate the behaviour of the four plastics when falling upon an identical object, signifying the differences in rigidity, bending moment, elasticity and plasticity.

Each figure shows the sheets falling under the effects of gravity (photos A and B), colliding with the polyhedra (photo C), settling over the object (photos D and E) before coming to rest (photo F).

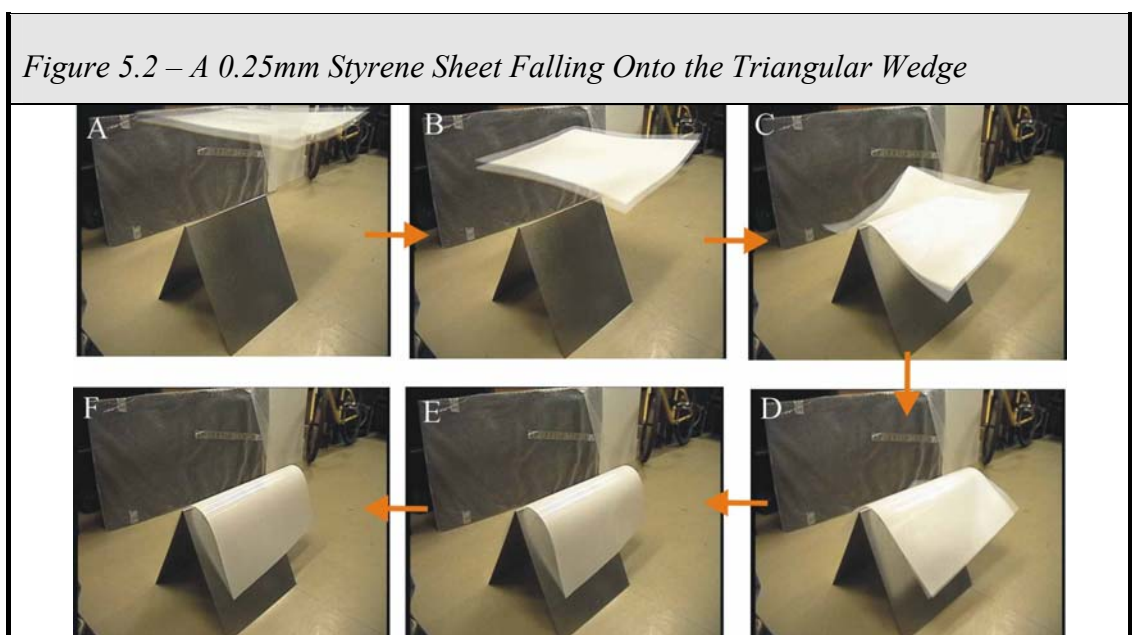
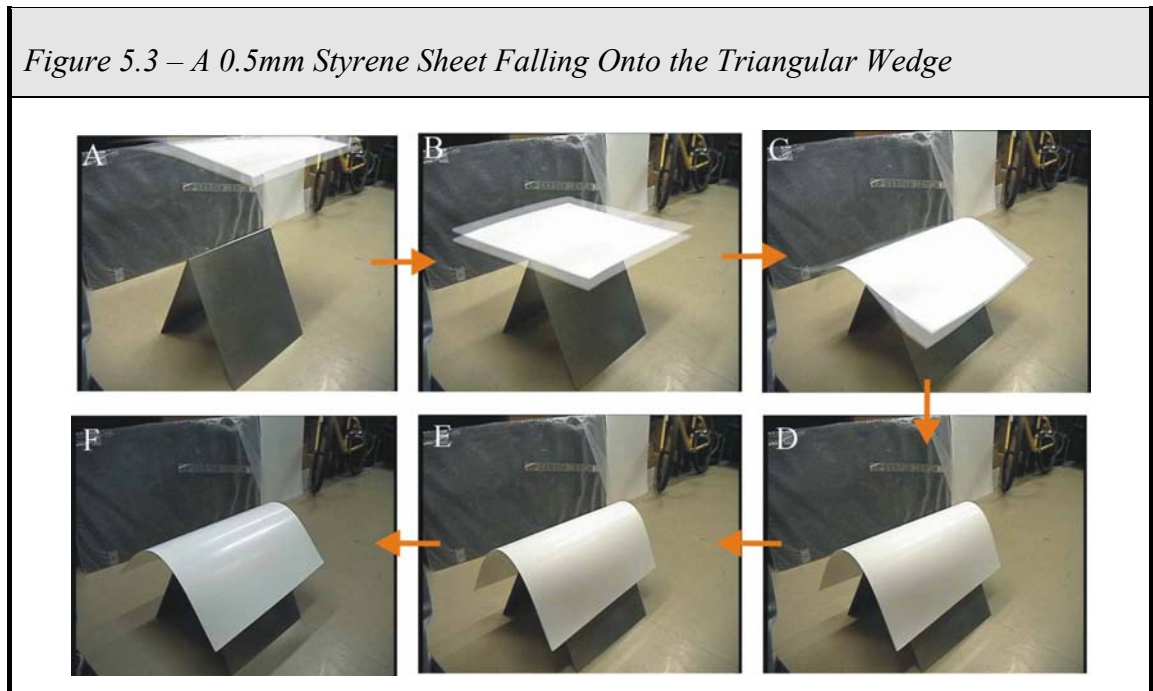
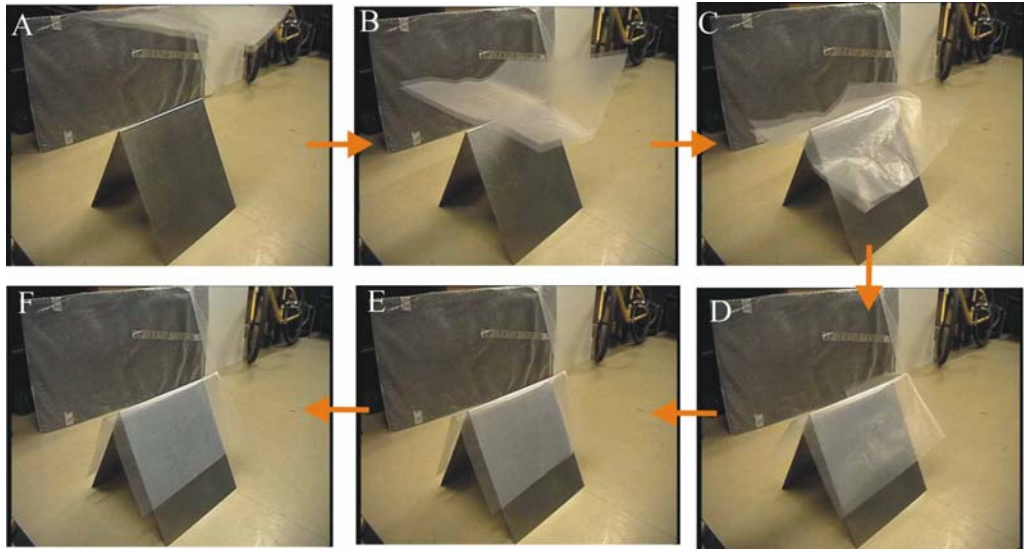


Figure 5.2 shows the 0.25 styrene sheet deflecting as it experiences the effects of air resistance while falling towards the object (photos A and B). However due to the flexibility of the thin 0.25mm styrene, the sheet deforms freely once it encounters the leading edge of the wedge (see photos C and D) before coming to rest (photos E and F).



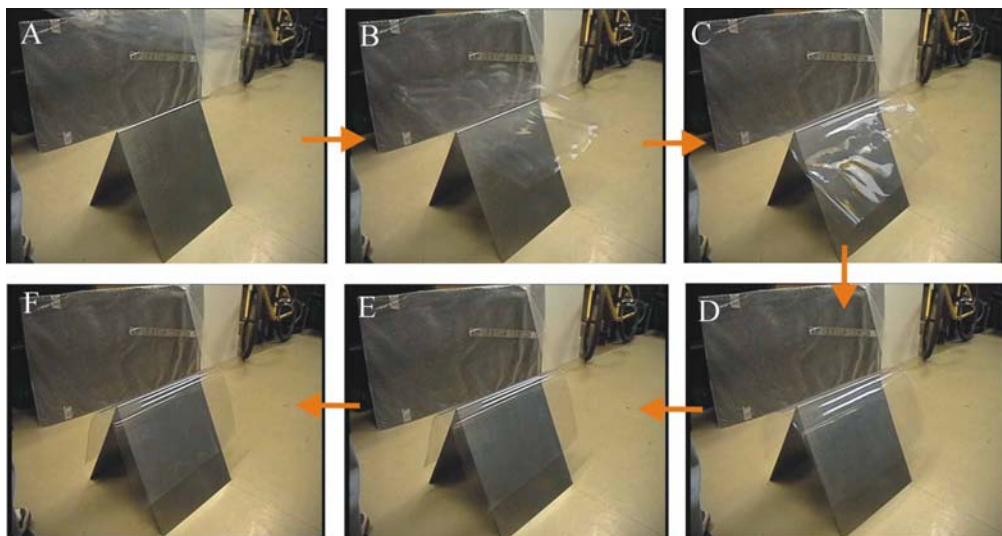
Due to its increased weight and inter-plane stiffness, the thicker styrene sheet is more able to withstand effects of air resistance as it falls under gravity (see figure 5.3, photos A and B). As the object collides with the leading edge of the wedge, the effects of an increased level of rotational stiffness results in the sheet having a wider curvature as it settles over the top edge of the wedge (photos D and E). An additional difference resulting from the greater thickness of the sheet are the stronger elastic forces that result in the laminate attempting to straighten itself once it has come to rest (photo F).

Figure 5.4 – An Industrial Grade Sheet Falling Onto the Triangular Wedge



The industrial grade PVC (see figure 5.4) is significantly thinner and less dense than the standard grade PVC. Hence, as the sheet falls under the effects of gravity (photos A and B) it is affected by the aerodynamic effects of air resistance, causing the edges of the sheet to bend upwards. Once in contact with the triangular wedge, its high level of flexibility causes the sheet to settle over the polyhedron with little residual curvature of the sheet across the leading edge of the wedge (photos C to F).

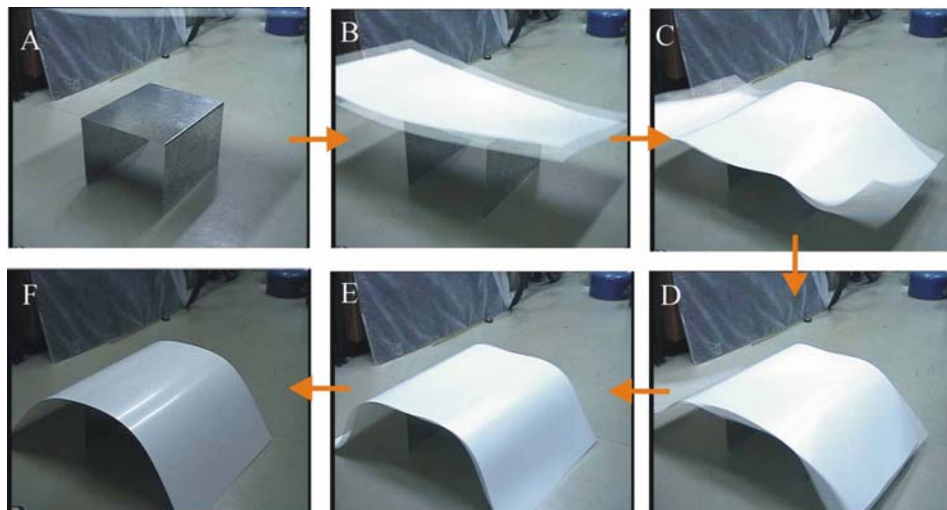
Figure 5.5 – A 0.4mm PVC Sheet Falling Onto the Triangular Wedge



The PVC sheet, made of a thick (0.4 mm) layer of plastic, behaves very differently from its 0.5mm styrene counterpart. The properties of the plastic (higher density and mass) reduce the effect of air resistance (photos A and B). Due to a lack of any significant bending stiffness, the sheet drapes freely over the object (photos D and E) with little evidence of it attempting to straightening itself (photo F).

Figure 5.6 demonstrates the behaviour of the 0.5mm styrene sheet as it collides with a cube-like structure. As with the triangular wedge case, the sheet retains its shape as it falls under the effects of gravity (photos A and B) and, following the collision, wraps itself over the cube (photos C, D and E). However, as it comes to rest, the moments within the sheet cause it to flex over the cube with only the edges of the upper cube surface retaining any contact with the sheet (photo F).

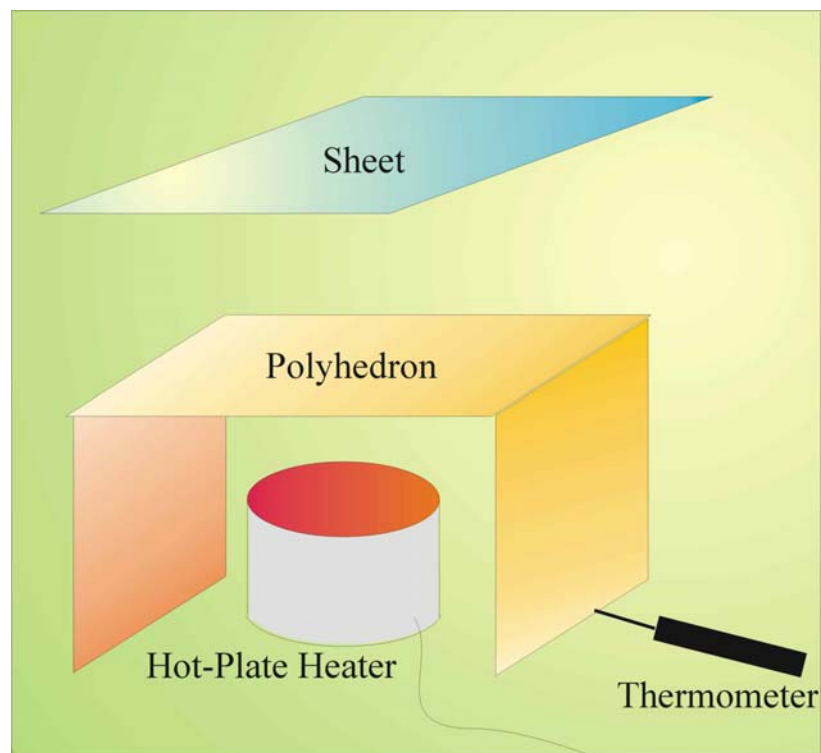
Figure 5.6 – A 0.5mm Styrene Sheet Falling Onto the Partial Cube



5.2.4 The Heat-Based Experimental Work

Following the conventional room temperature experiments, testing was conducted into how the behaviour of the sheets would differ when encountering heat. Due to safety restrictions, and a lack of suitable testing facilities, the heat-investigation was conducted using an electric hot-plate heater inserted underneath each of the test polyhedra (see figure 5.7). A digital thermometer was used to record the temperature of the lower edge of the polyhedron before, during and after the testing process.

Figure 5.7 – The Heat Experiment Testing



A digital video camera was used to film the sheets falling and subsequently melting onto and over the objects.

In total three shapes were used for the heat testing experimental work, these being the Triangular Wedge, the Half Cylinder and the Upright Cylinder. It was initially intended to test all four sheets for their heat sensitive properties. However, due to insufficient

facilities for safely distributing the necessary levels of heat to produce visible evidence of heat deformation it was deemed expedient not to test the thicker styrene sheet on the triangular wedge sheet. Also, to aid brevity, the results of the material from the half-cylinder testing have been omitted from these results as it was found that the sheet performed in a very similar manner to the 0.25mm styrene sheets but with reduced levels of deformation due to the thickness of the sheets. Due to their similar deformation patterns to the 0.25mm styrene and 0.4mm PVC, with the exception of the upright cylinder tests, the 0.5mm styrene and industrial PVC results have also been omitted from this chapter section.

Triangular Wedge Heat Testing

Once the 0.25 styrene sheet settled over the wedge object (photos A and B in figure 5.8), it took a further minute before minor creasing started to occur along the sheet overhangs, (photo C) with further warping starting to appear after 3 minutes (photo D). Photo E highlights the increased definition of the sheet across the leading edge of the object as the plastic starts to distort plastically after 6 minutes with definite warping along the wedge faces after 9.5 minutes (photo F). This becomes more pronounced after 12 (photo G) and 15 (photo H) minutes prior to warping along the front and side of the wedge after 15 minutes (photo I) followed by bending and creasing at 16 and 19 minutes (photos J and K). After 20 minutes, the sheet overhang has bent over the front of the wedge (photo L) which becomes crumpled up after a further 2 minutes (photo M) before starting to bend downwards and enclosing the front wedge after 26 minutes (photos N and O).

Figure 5.8 – A 0.25mm Styrene Sheet Melting over the Triangular Wedge

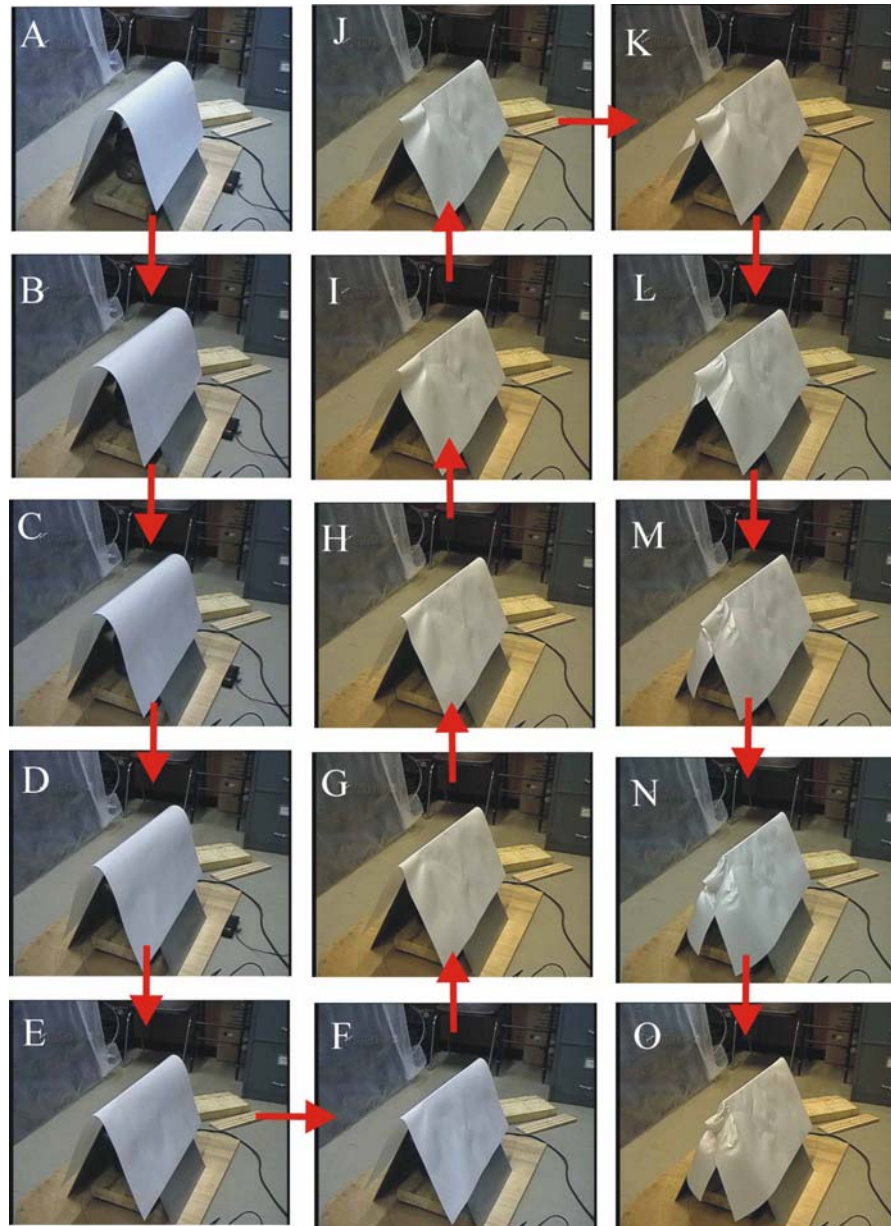
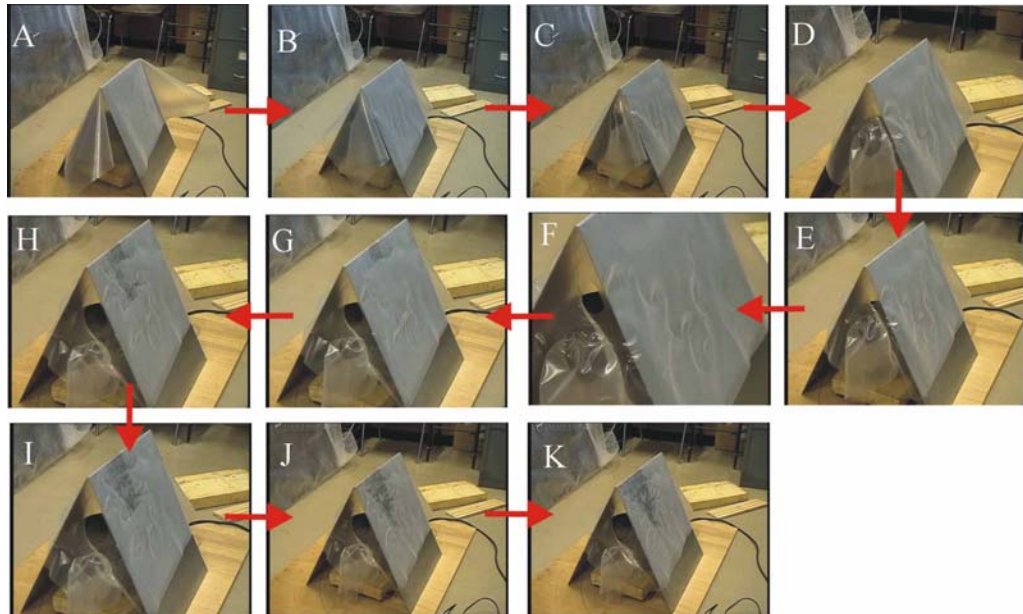


Figure 5.9 shows how the industrial PVC plastic behaves when placed on the heated triangular wedge (photo A). After 14 seconds, the overhang folds protruding along the front and back of the sheet weaken and drape vertically down the front of the wedge (photo B) before the initial signs of melting along the front and across the faces of the object between 24 (photo C) and 44 seconds (photo D). Separation across the front of the sheet occurs after 54 seconds (photo E) with the newly formed hole widening over the next 10 seconds (photo F). The expansion of the hole and melting across the wedge

face occurs between 1.5, 2.5 and 3 minutes (photos G, H and I) before the initial formation of a large liquid-like patch across the wedge face after 4 minutes (photo J) which gradually encompasses the entire face over the remaining 2 minutes of the experiment (photo K).

Figure 5.9 – An Industrial Grade Sheet Melting Over the Triangular Wedge



The underlying purpose behind these experiments was to produce a tangible benchmark against which the aesthetic capability of the proposed model can be tested. The next stage of this evaluation process was to conduct experimental work with the simulation system to provide comparative appraisal of the suitability of the model to real world modelling. This not only produces tangible results as to the accuracy of the system, but also aid the calibration process of the environment.

5.2.5 Simulation-Based Experimental Work

Similar to the physical experiments, the simulation work is divided into two specific categories, the first set of trials focusing upon room temperature sheet behaviour while the second group of investigations analyse the effectiveness of the MaSSE thermal softening/virtual heat system.

Room Temperature Testing

Room temperature testing focussed principally upon the behaviour of the virtual sheet as it falls under the influences of gravity and interacts with polyhedra. To achieve this, a number of criteria are used for the basis of the examination upon which the effectiveness of the system will be judged.

These markers include:

1. The bending moment induced in the sheet as it interacts with the polyhedron.
2. The draping pattern of the sheet as it falls over the object.
3. The level of realism portrayed by the sheet as it reacts to its environment.
4. The overall stability of the simulation.
5. The speed performance of the model.

The results of these parameters are then used to judge the overall suitability of the proposed methodology to modelling room temperature sheet behaviour.

Thermal Softening/Virtual Heat Testing

The thermal softening system is comprised of two different approaches to simulating the effect of radiation on the sheet. These are, the Oven Thermal Softening and Room-Heater (radiation based) systems which are individually examined in the testing cycle.

- **Oven Thermal Softening System**

The oven thermal softening system represents the more basic of the two approaches which, as its name suggests, simply saturates the entire sheet with a evenly distributed heat, similar to that experienced in an oven.

This approach is simpler than an authentically accurate oven heat thermodynamic simulation since it ignores the effects of thermal conductivity (i.e. heat being passed through the sheet itself) and the effects of the heat as it moves around the room (i.e. convections).

The purpose of this virtual heat model was to examine the effectiveness of reproducing heat exposure upon a mass and spring based sheet system, without the need for complex calculations and algorithms which may have affected the overall look of the deformed sheet. The model also aims to assess the suitability of three-dimensional mass-spring networks at reproducing the visual appearance of plastic sheets as they melt.

- **Room-Heater (Radiation) System**

The virtual heater model was developed to introduce higher temperatures into the environment with greater levels of realism than the oven thermal softening approach, while adhering to the fundamental principles of the research by avoiding the use of complex thermodynamic equations to reproduce the synthetic radiation as it is transferred through the room.

The suitability of this system will be judged upon a number of criteria, which include:

1. The suitability of directional heat sources to spring-based modelling.
2. Determining whether the pseudo-conductive heat algorithm provides the heater system with any additional level of realism.
3. Whether the melting pattern of directional heat enhances the visual experience of the simulation.

5.3 Aurora System Testing

5.3.1 Test Plan

The initial testing focused on the room temperature experimentation, studying the behaviour of the sheets as it underwent numerous collisions with foreign bodies of different size, shape and spacing. Some of these were similar to the real world testing studied in section 5.2 so that direct comparisons may be made between the simulation and its real world equivalent.

The behaviour of the three calculation engines will then be examined along with each technique's suitability to performing real world simulations in terms of realism, stability and performance. Each of the three sheet meshes will then be evaluated to ascertain which design provides the most effective model for simulating the behaviour of the sheet. The system shall then be calibrated to determine which settings provide the most realistic results, together with comments based upon the observed results.

5.3.2 Simulation Test Equipment

The work presented in this chapter was conducted on an IBM Thinkpad 1.4 Ghz Intel Pentium M Centrino laptop computer running Windows XP Professional equipped with 768MB RAM and a 32MB ATI Mobility Radeon 7500 graphics card.

The application is written and compiled using Microsoft Visual C++ release 6 (service pack 5) with the Microsoft implementation of the OpenGL library (release 1.1), the GLuT toolkit and the GLUi OpenGL interface system.

5.3.3 Test Schedule

In order to test various capabilities of the Aurora sheet simulation system, a default setup was required that allowed variations in parameters to be measured. To aid this, all simulations encompassed the following settings unless otherwise specified.

<i>Table 5.1 – Simulation Testing Parameters</i>	
Parameter	Value
MaSSE Engine	2 nd Order (Euler-Richardson)
Time-step	0.20 (Seconds)
Sheet Structure	Triangular Mesh Matrix (TRiMM)
Sheet Size	11 wide, 2 deep, 2 long
Restitution Coefficient	0.1 (10%)
Kinetic Friction Coefficient	0.3
Air Friction (Damping Coefficient)	0.97 (3%)
Sheet Display Settings	Display top sheet layer; Display Springs; Do not display masses
NURBS	Engaged
Heat System	Off

5.3.4 Results of Sheet Dynamics Testing

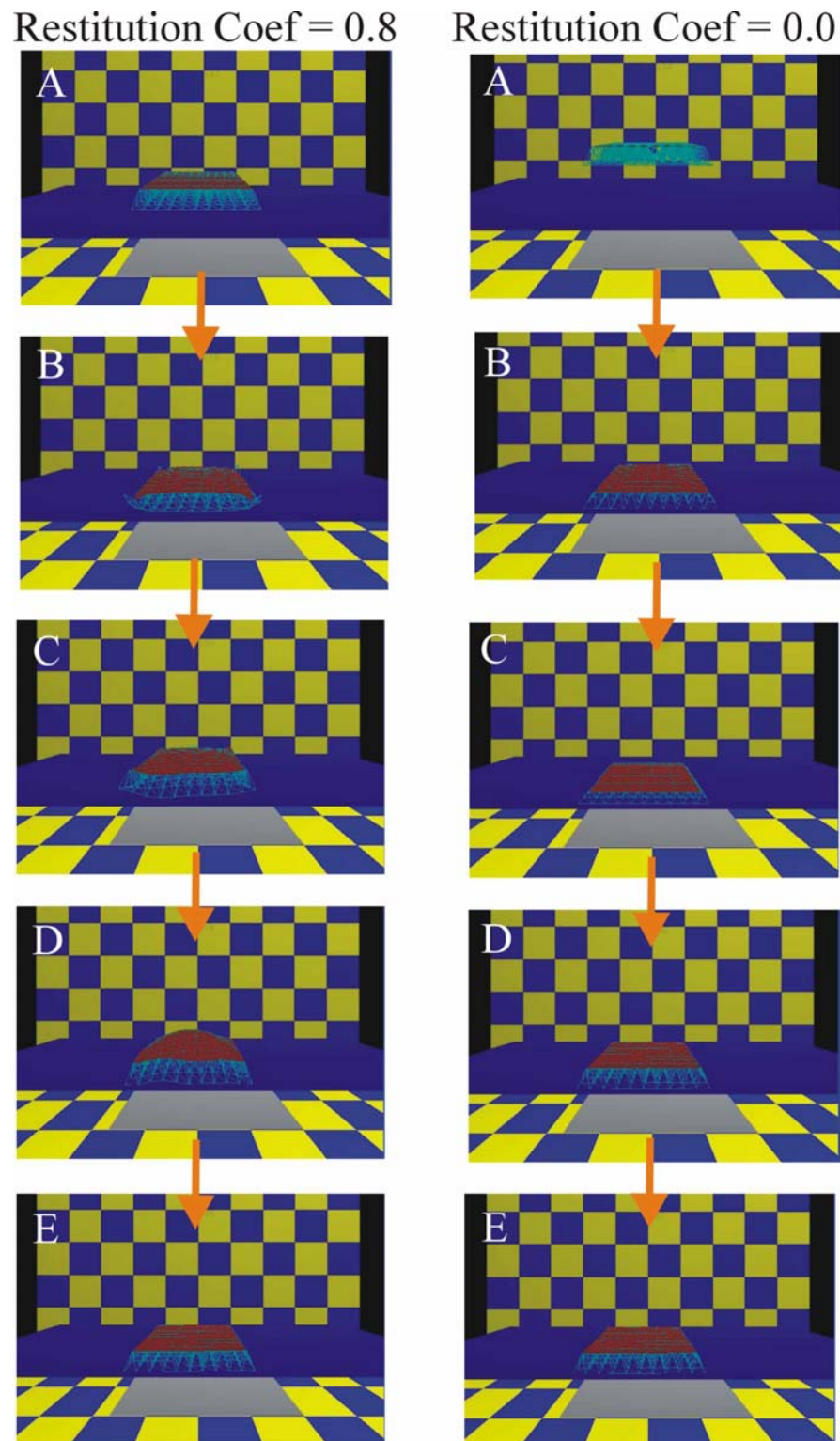
Restitution Coefficient

When a restitution coefficient is applied to the dynamics model (see figure 5.10, left column), the level of energy present within the model causes the outer masses to move along the collision plane (print B) before causing the whole model to symmetrically lift off the collision plane (prints C and D) before coming to rest once more along the polyhedron's surface.

The behaviour of the masses does vary noticeably once the restitution coefficient is zeroed, the sheet structure becomes more placid as sheet energy is dissipated through contact between the sheet and the surface (see figure 5.10, right column).

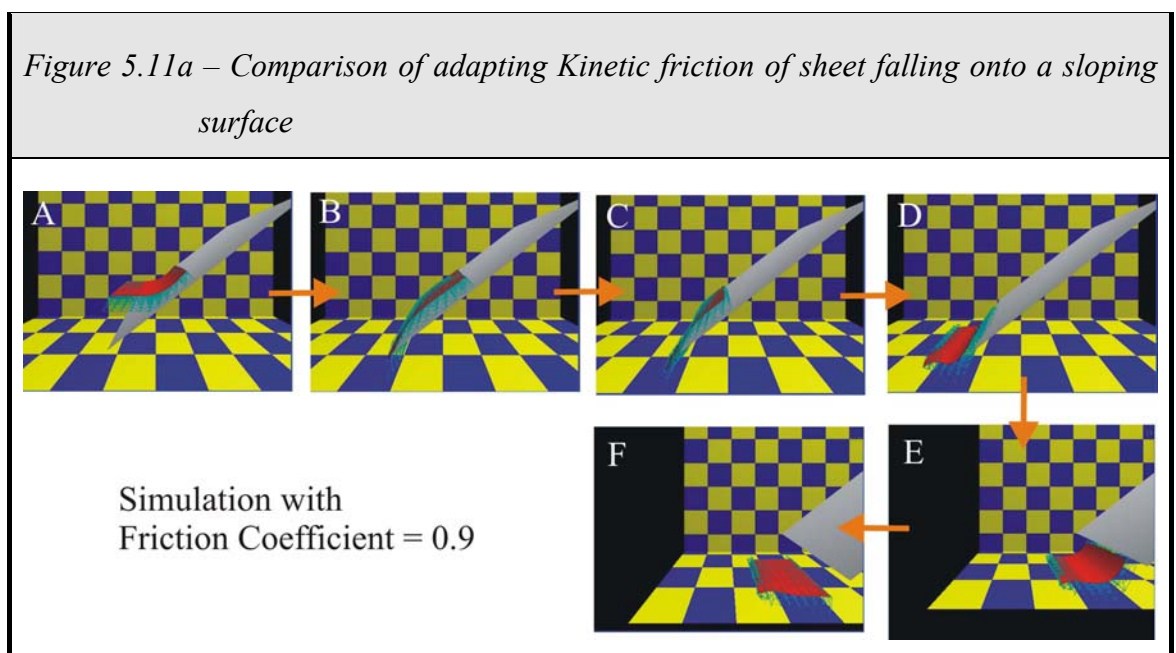
As the sheet collides with the surface (print B) the masses retain their lateral structure within the sheet, even when the interlayer springs form significant levels of compression forces upon the lower sheet layer (print C). When the upper sheet layer is restored to its resting position above the lower sheet, the sheet remains static above the surface without any 'bouncing' (prints D and E).

Figure 5.10 – Comparison of adapting Restitution Coefficient of sheet falling onto a flat surface



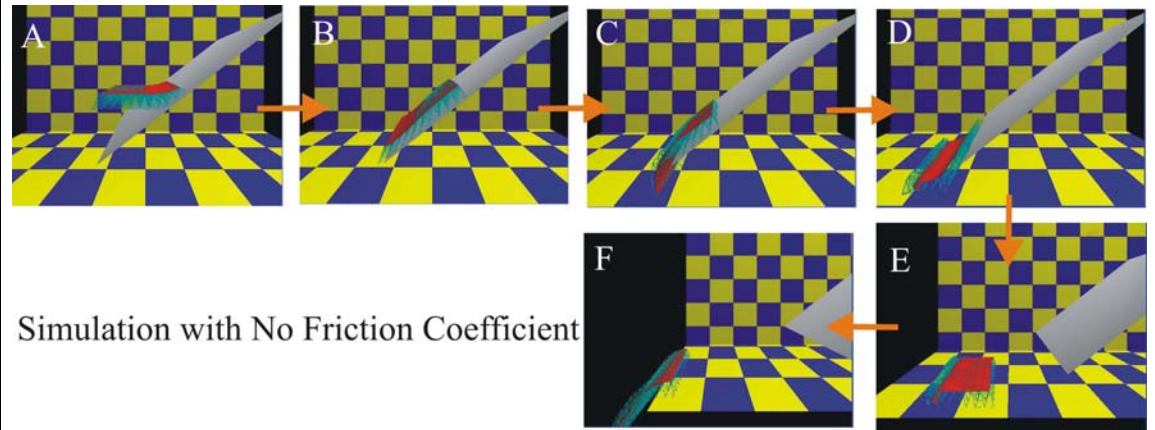
Kinetic Friction

When the simulation's kinetic friction coefficient is set to a high value (i.e. 0.9), the sheet (figure 5.11a) falls down a slope with a high angle of incidence with the ground, resulting in the sheet colliding with the surface (print A) before creeping down the polyhedron (prints C and D) until it reaches the flat ground (print E) where it comes to rest (print F).



However when no friction exists between the sheet and a surface (figure 5.11b), as the sheet comes into contact with the surface (print A), gravitational forces push the sheet along the face, gaining momentum as it does so (prints B and C). On reaching the environment floor the sheet continues to speed across the surface (print E) until eventually disappearing from the environment (print F).

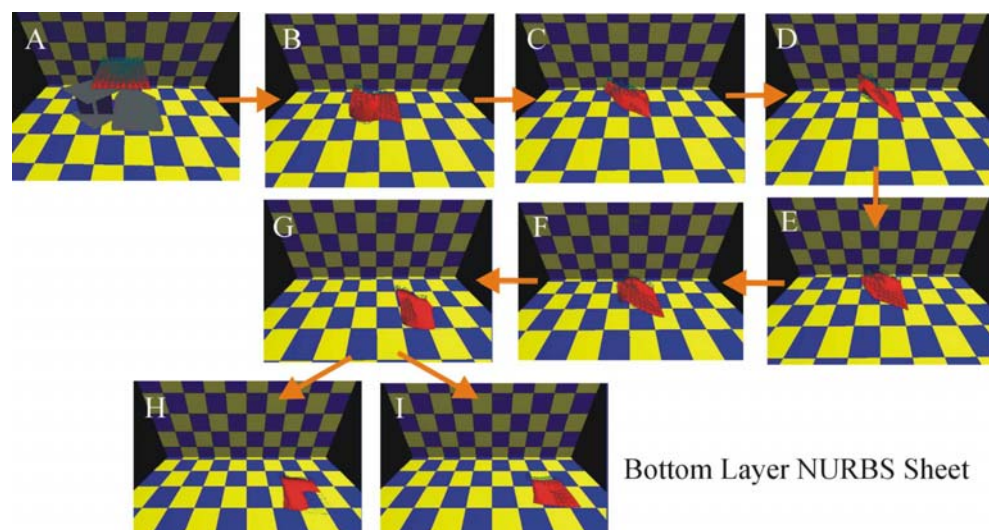
Figure 5.11b – Comparison of adapting Kinetic friction of sheet falling onto a sloping surface



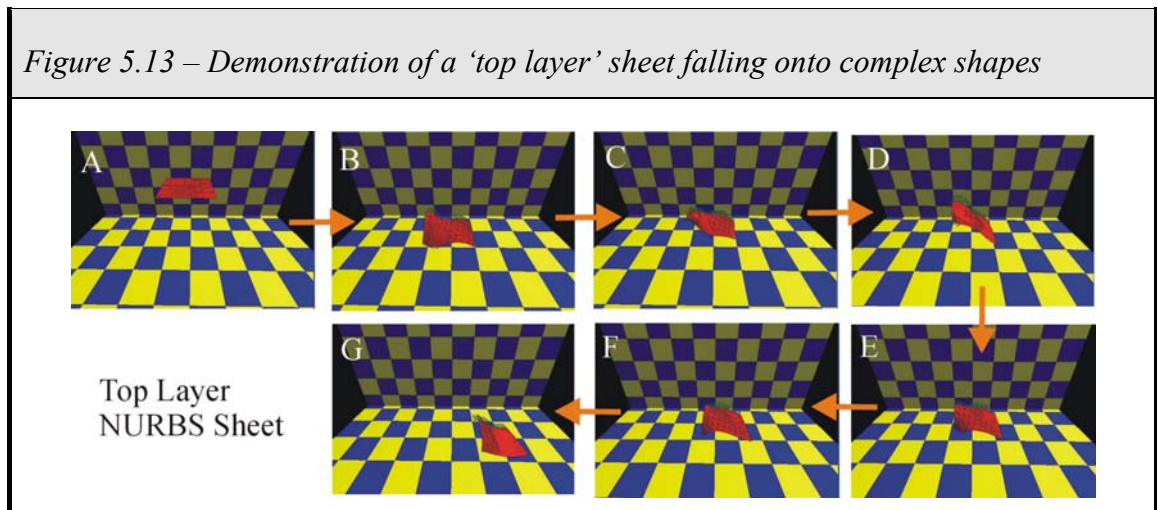
Sheet Layer Comparison

A question during the course of this study (Mahal et al, 1999) concerned any differences in the behaviour of the model when using the bottom layer of the sheet. Figures 5.12 and 5.13 illustrate how mapping the bottom layer fails to cause any significant variation in sheet behaviour over use of the top sheet.

Figure 5.12 – Demonstration of a 'bottom layer' sheet falling onto complex shapes



One identifiable issue, particularly when using the NURBS system was the tendency of the bottom layer to be rounded so that it sinks below any surface that the sheet comes into contact with. This is best illustrated in Figure 5.12 where print H represents the sheet with the bottom layer displayed (which the front right section of the sheet displayed under the floor surface) whilst print I presents the sheet with the upper sheet displayed (with the entire sheet displayed).



NURBS Comparison

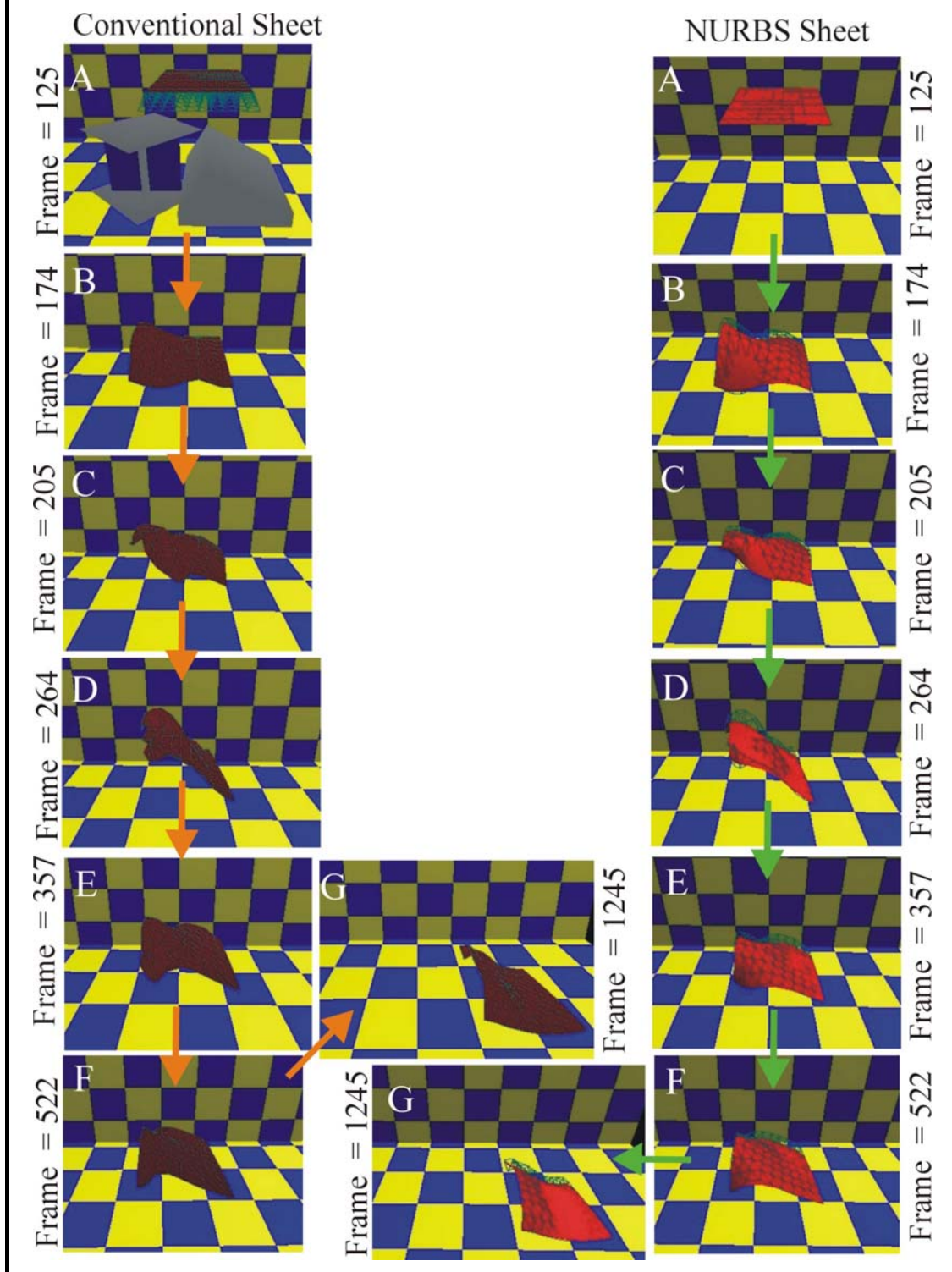
Use of the NURBS system does provide some significant improvements in the results of the sheet surface, particularly under conditions of high deformation. However the nature of smoothing algorithms such as the NURBS system provides inaccuracies in the visual appearance of objects in order to achieve their smoothing effect. For objects acting independently within free-space, this would have no detrimental effects upon a simulation. It is only when a NURBS model comes into close proximity with another object that the shortcomings of the approach become apparent. As no exact positions of the revised NURBS shape are produced by the system, it is impossible to determine where the revised positions of the Mass points are placed within the NURBS sheet. As such, the system is unable to provide any collision detection checking with the NURBS sheet. Instead the collision detection routine must continue to operate with the conventional sheet points, which can often lead to parts of the sheet partially sinking into the simulation polyhedron. An attempt to overcome this issue would involve

developing a bespoke NURBS routine that took in the mass positions, performed a NURBS surface calculation, then subdividing the area of the new surface into a grid structure (based upon the dimension of the original sheet) before determining the displacements of the revised masses. This would have significant implications upon system performance as any bespoke NURBS routine would be unable to take full advantage of the optimised OpenGL routines and any backward tracing of the revised Mass displacements for each time-step would compound this further.

Figure 5.14 illustrates the differences between the conventional sheet (left column) and NURBS sheet (right column). To eliminate the effects of the sheet ‘sinking’ effect, only the top sheet behaviour with the top-layer springs is displayed.

Due to the granularity of the sheet mesh (i.e. the distance between masses), the conventional sheet edge appears ragged, particularly during the early stages of deformation with the sheet (prints B through to E), which is clearly evident when compared to those of the smoothed sheet. The NURBS shading algorithm provided by the OpenGL also enhances the visibility and look of the sheet (prints D, E and F) which aids the user in visualising the sheet behaviour.

Figure 5.14 – Sheet Simulation comparison of NURBS functionality



Spring Stiffness behaviour comparison

One of the key aspects of this research has been the development of a realistic sheet structure while maintaining a stable simulation. To achieve this, a number of key parameters must be set which significantly influence the behaviour of the sheet. These principal attributes include spring stiffness, mass value, sheet damping, simulation time-step and integration method (i.e. MaSSE engine selection). The variables must be linked together in order to form a coherent and stable simulation. Examples include the time-step of a stable simulation being dependent upon the choice of integration method, and increasing the spring stiffness of the sheet should also result in either the increase of the particle mass-values and/or reduction of sheet damping. If these relationships are ignored, then adapting one parameter in isolation would result in an unstable simulation which, as it approaches numerical instability, would cause the sheet to explode, with masses flying apart in all directions.

To adapt the spring stiffness of the sheet while producing a comparable simulation, the decision was taken to reduce the sheet damping whilst maintaining the mass of the particles and thus the entire sheet. Experimental testing (figure 5.15) has demonstrated that, when weak spring stiffness are used, they form a highly elastic surface that extends excessively when coming into contact with foreign objects. This model tends to gain excessive levels of energy as the system attempts to restore its shape. To counteract this and retain stability, the system is capable of accommodating higher levels of damping within the sheet. If the spring stiffness is set too low (figure 5.15, left column), the sheet fails to restore its original shape. However a marginal increase in spring stiffness results in the sheet acting like an elasticised model with, after their initial extension, the springs contracting with such ferocity that it causes the sheet to spring back upon itself, resulting in the mesh structure becoming interlinked with itself. An alternative outcome is the introduction of excess energy into the system following the spring contraction resulting in simulation instability.

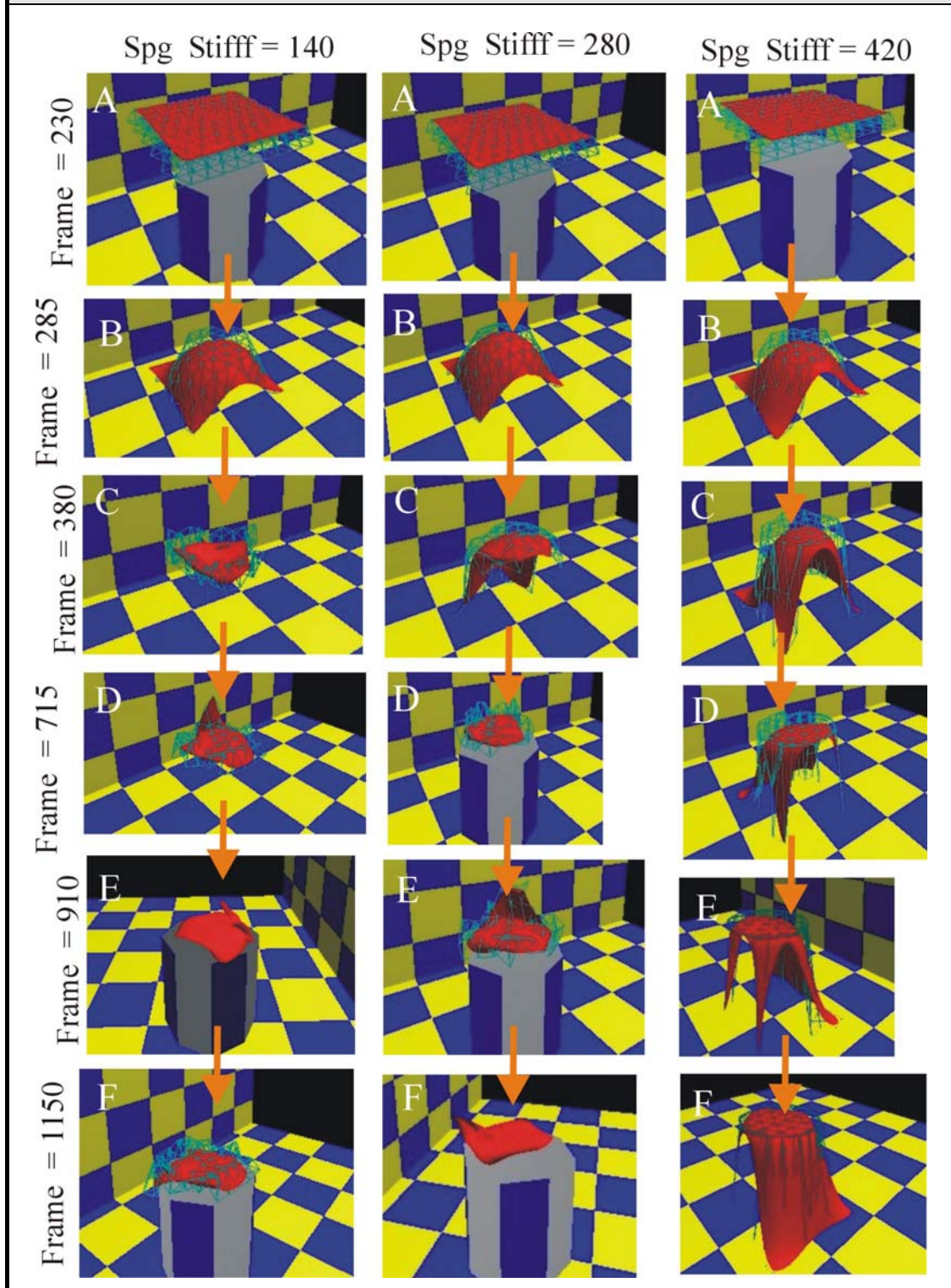
As the spring stiffness is increased to compensate for this excessive level of deformation during contact scenarios (figure 5.15, centre column), the model is less capable of enduring higher levels of sheet damping, which must therefore be reduced.

When further enlarged (figure 5.15, right column), the spring stiffness then, rather than the stiffening up causing a mildly elastic structure, the sheet becomes highly plastic with the structure failing to attempt to restore its original form.

Attempts to increase the level of bending stiffness through a reduction of the mass levels within the sheet often result in the complete instability of the model if reduced beyond defined levels (for example 0.4g with spring stiffness of 400g/mm). This behaviour was unexpected as it had been predicted that attempts at increasing the spring stiffness without a reciprocal increase in mass would result in greater levels of bending moment.

Unusually the simulation demonstrated in figure 5.15 illustrates a lower level of deformation in the weaker sheet stiffness's due to the weaker models flexing excessively upon contact with the polyhedron which resulted in it folding upon itself on the top (horizontal) surface top.

Figure 5.15 – A Comparison of Spring Stiffness Variation Upon Sheet Behaviour



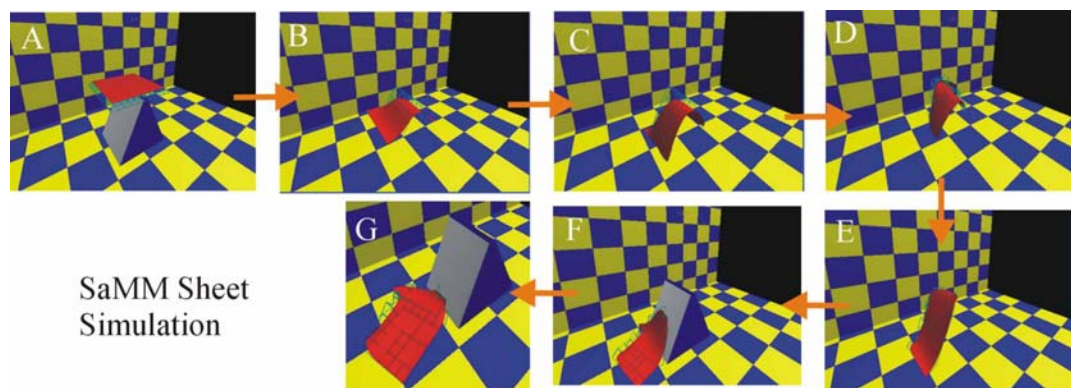
Sheet Structure Comparison

One of the early investigations of the research involved studying the effects of varying sheet structures upon the overall behaviour of the laminate as it fell and interacted with its environment. This involved analysing the behaviour of each sheet under room temperature conditions and under heated conditions.

The SaMM model formed the simplest and weakest structure which, under normal conditions, resulting in a system that was more likely to become unstable when encountering high contact forces with other objects. Even when under more gentle loads (figure 5.16), the matrix had greater levels of susceptibility to deform and stretch than its counterpart structures. Due to the relatively small reduction in the number of springs per sheet (in relation to the total numbers in any of the sheet structures) little performance gains were achievable using these models over the other shapes.

However, this relative weakness proved to be its greatest strength when modelling thermal softening as the shape allowed the effects of modest levels of virtual heat to have a greater influence upon the sheet shape.

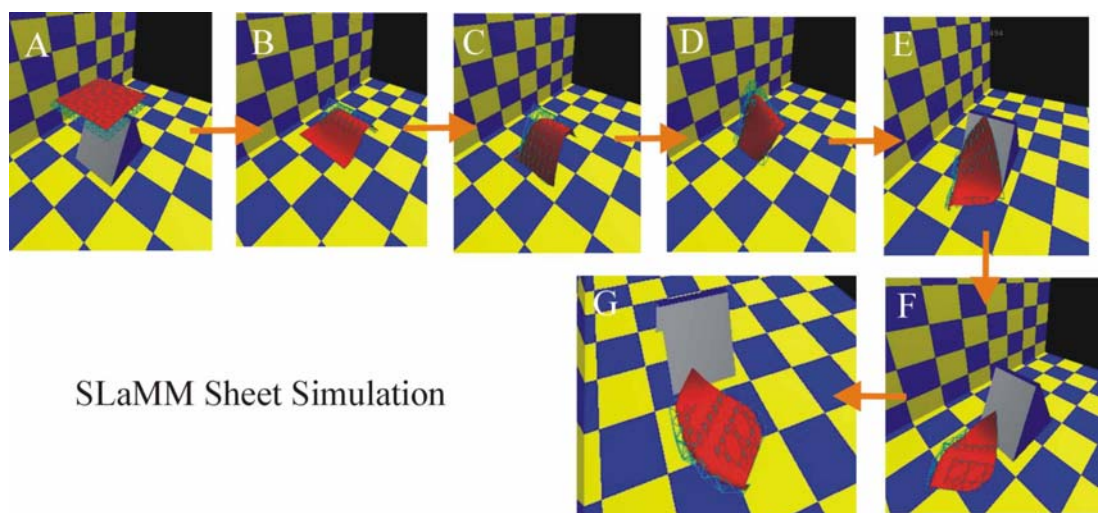
Figure 5.16 – A SaMM sheet falling upon a Triangular Wedge shape



As expected, the SLaMM structure formed a stronger unit than the SaMM. However, the expectation had been that due to its triangulation cross-member across each quadrant, these sheets would perform significantly better than those of the non-triangulated SaMM model. This was not the case and instead the use of a single cross-member across one side of each quadrant across the entire sheet led to minor distortions in the model's behaviour resulting in non-uniform deformation across the sheet surface. Even sliding down a surface would cause the sheet to yaw across its Z-axis (see figure 5.17), when an identical SaMM (see figure 5.16) would not. Similar results were achieved when exposing the SLaMM structures to virtual heat, resulting in a diagonal distortion across the laminar surface, running in the opposite direction to the cross-member flow across the sheet quadrants.

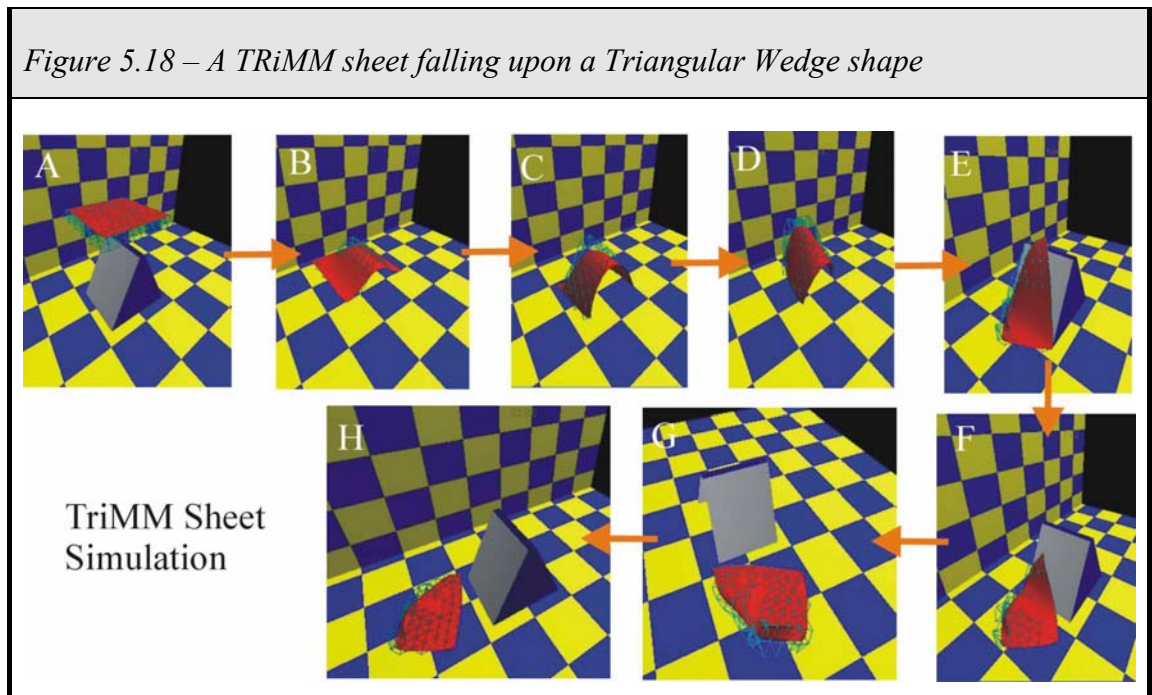
Following extensive experimentation, it was felt that the use of a second cross-member would produce significantly enhanced results through the creation of a stronger and more uniform structure. An alternative approach would extend this idea further through the use of additional cross member springs that interlink masses that are one-removed from each other and so 'jump' over their adjoining masses points. This concept is examined further in the Discussion chapter.

Figure 5.17 – A SLaMM sheet falling upon a Triangular Wedge shape



The TRiMM structure formed the most robust method with sheets deforming in a more regular manner than the 2 Square matrix methods. Its extensive use of triangles resulted in a structure that benefited from greater levels of stability, sheet behaviour and rigidity when under gentle loading (figure 5.18). The differences in performance between the SLaMM and TRiMM systems were unexpected. However, most of this is attributable to the poor performance of the SLaMM model. Having said this, even with a second member cross-member installed into the SLaMM structure, it would not be expected to perform as well as the TRiMM structure.

When exposed to the effects of virtual heat, the strength of the structure resulted in greater levels of exposure before the effects of the spring weakening was able to take hold in melting the structure.



Stiff Sheet Shape Collisions

As detailed in the Spring Stiffness Behaviour Comparison, the use of high levels of spring stiffness results in unexpected plastic behaviour in the sheet when exposed to high contact forces. Figures 5.19, 5.20 and 5.21 demonstrate this unusual behaviour. This reaction is caused due to the high levels of sheet damping that are required to

prevent excess energy build-up and so retain sheet stability during high-velocity impact simulations. Once the sheet reaches its maximum elastic formation, the sheet enters into an over-damped state that prevents the internal springs from contracting and thus the sheet from re-obtaining its elastic structure.

Paradoxically, figures 5.19 and 5.20 highlight the clear similarities between the highly strung sheets and those of an elastic sheet made up of weak-structured elastic material such as PVC. The simulation behaviour mimics how when in contact with the cube (figure 5.19), the simulation closely resembles the behaviour of the 0.25mm Styrene and Packaging grade PVC while the non-NURBS based simulation of the wedge (figure 5.20) provide a close resemblance to the 0.4mm and Packaging PVC.

Figure 5.19 – Comparison of stiff and Real world Sheet Behaviour over a Cube.

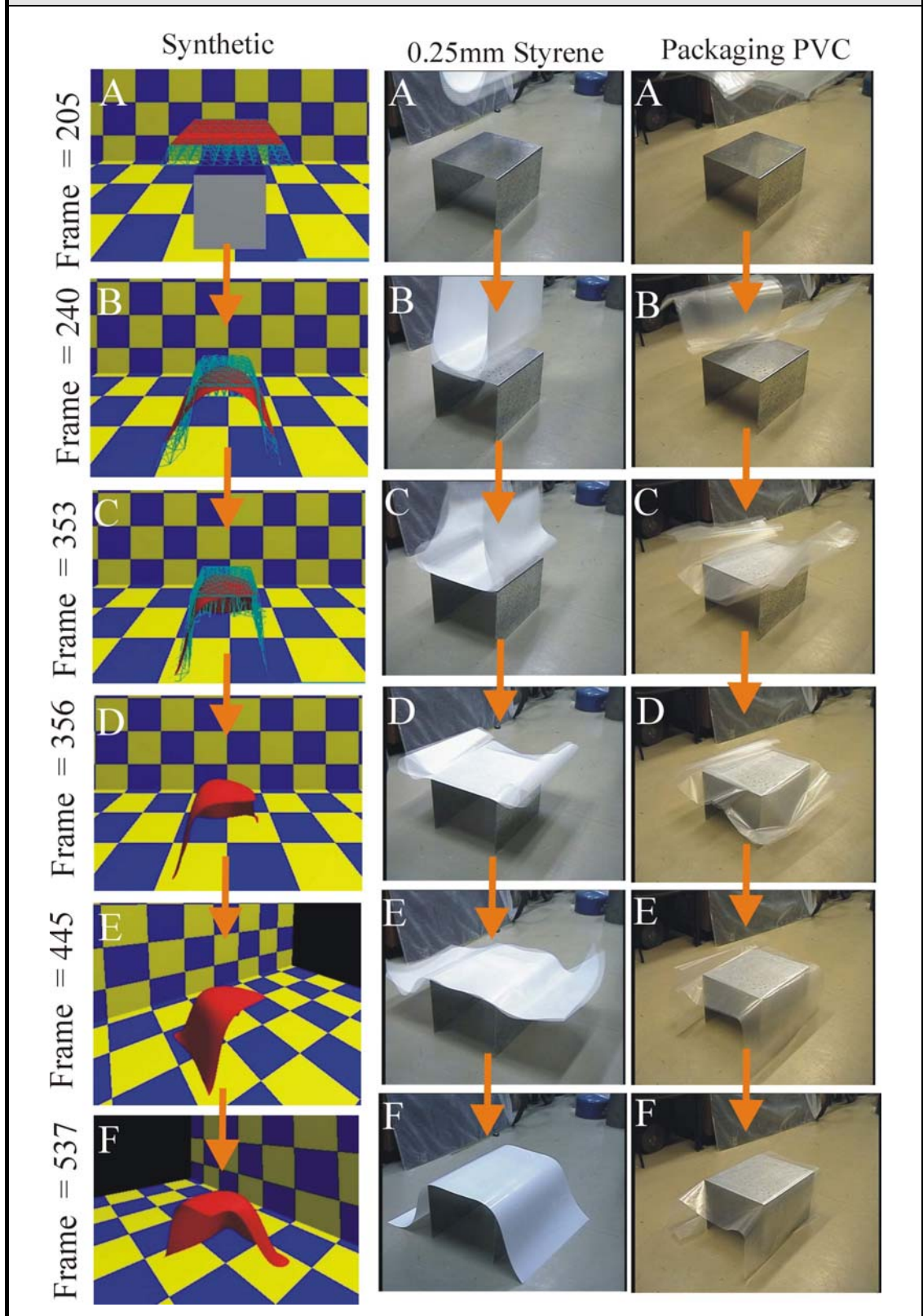


Figure 5.20 – Comparison of Stiff and Real world Sheet Behaviour over a Wedge

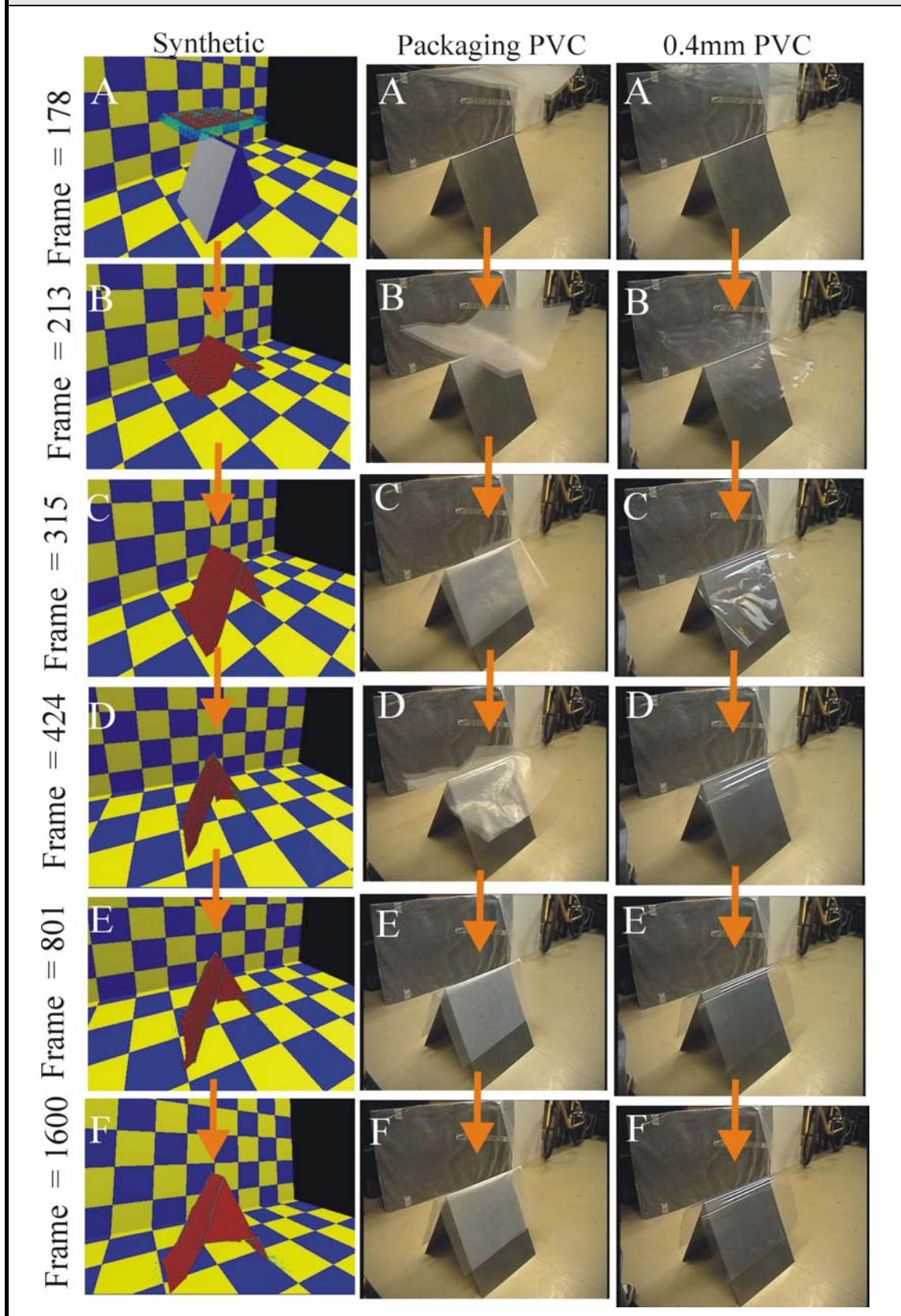
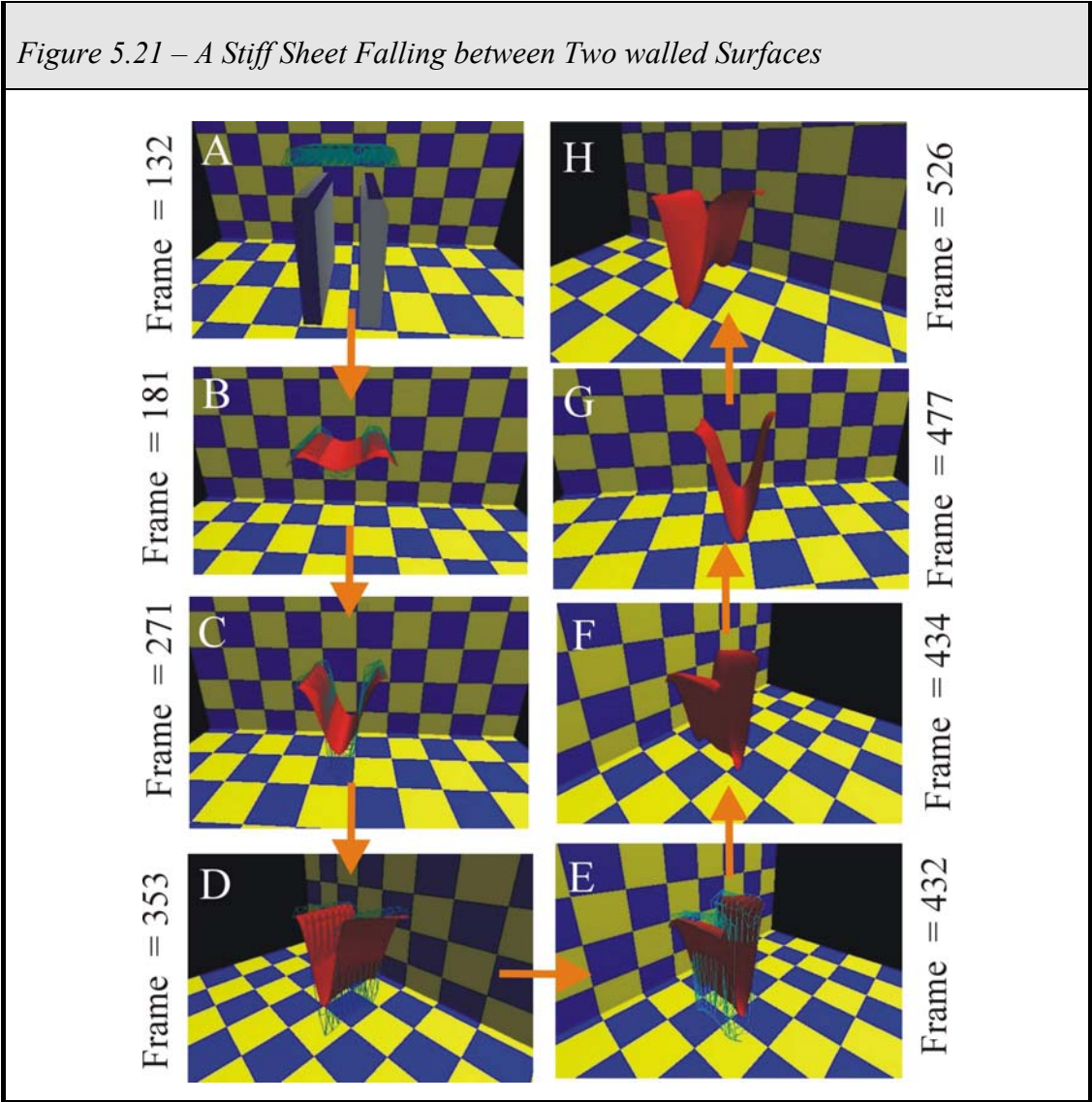
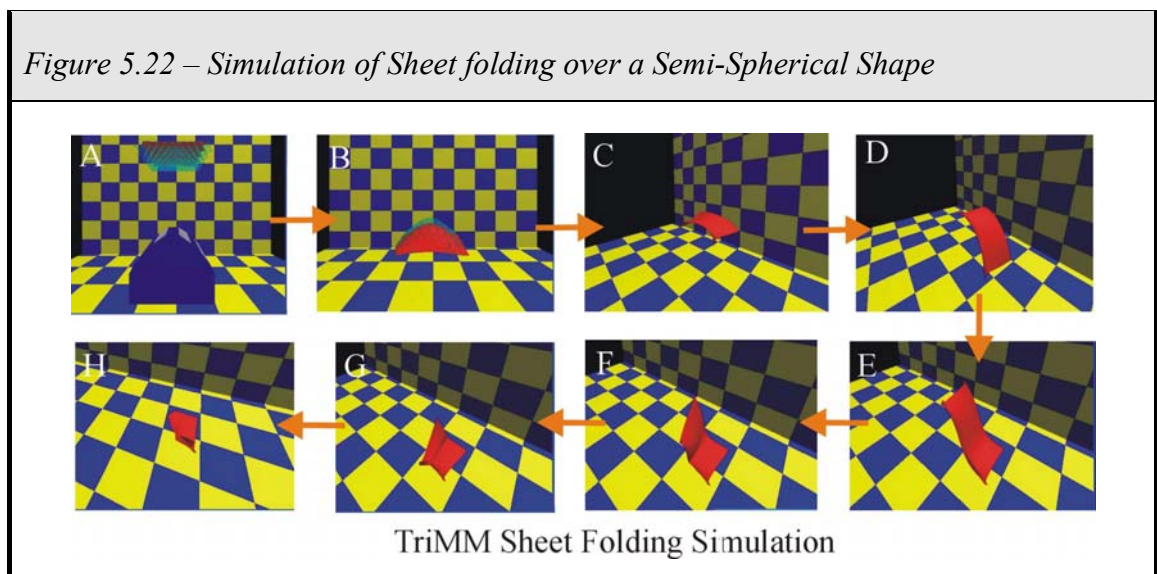


Figure 5.21 illustrates the flexible nature of the stiffened sheets as they come to rest between the two walls without any of the spring back that accompanies any conventional simulations with reduced levels of spring stiffness. Prints B through to E highlight the true position of the mass positions laying over the walled surfaces prior to the sheet smoothing by the NURBS algorithm.



Sheet Behaviour

Figure 5.22 presents a conventional synthetic sheet as it falls over a hemi-spherical polyhedron (print A), stretching over the surface (print B) before recompressing itself into its unloaded shape and sliding down along the surface (prints C, D and E). It eventually folds over itself (prints F and G) before coming to rest in its folded state (print H).



Sheet Softening and Melting Using the Oven Thermal Softening System

The oven thermal softening system allows a sheet to uniformly deform over an object by weakening the sheet springs until a pre-defined value has been achieved. This value is dependent upon a number of simulation parameters (e.g. time-step, mass value, spring stiffness, MaSSE engine etc.) to ensure that the spring stiffness value is not over-weakened to retain simulation stability.

Figure 5.23 illustrates a typical sheet melting scenario whereby a SLaMM sheet is dropped onto a polyhedron (print A) which it then drapes over and stretches over (print B), before contracting (print C) and settling (print D). The oven thermal softening system is then activated (with a temperature value of 21 degrees) causing the sheet ends to slowly extend over the polyhedron (prints E and F) with the sheet's proportions

adjusting, moving from a square shape and instead adopting a rectangular appearance (print G). As the sheet deforms further, the elongation along the x-axis extends while contacting the surface along the z-axis (prints H and I), thus ensuring that the surface area of the sheet does not grow substantially over its original value.

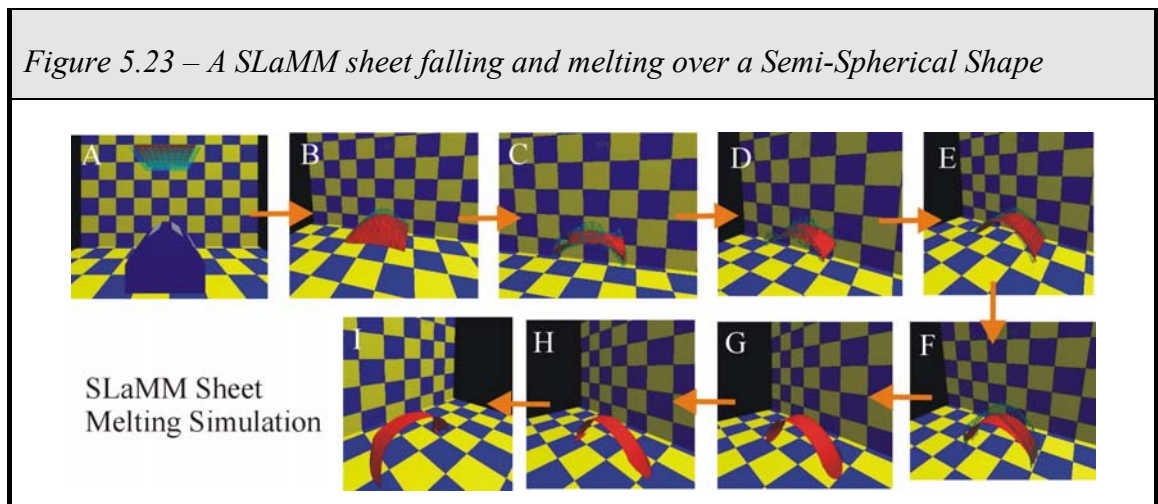
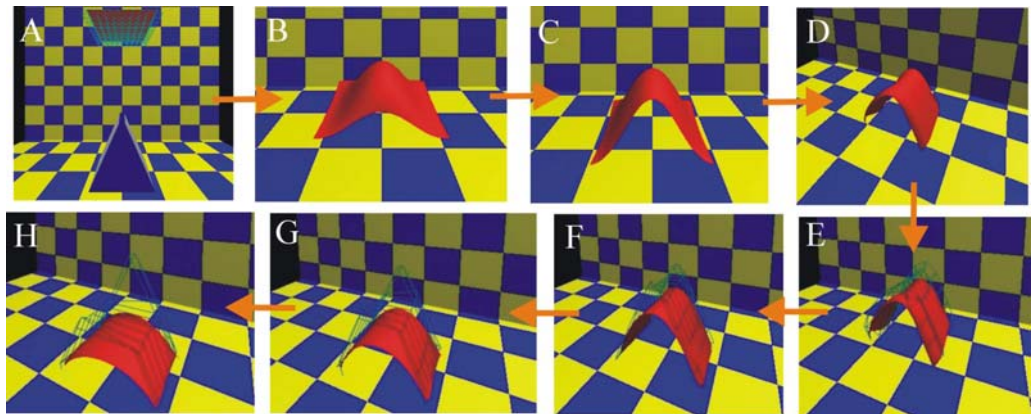


Figure 5.24 illustrates the deformation of a sheet as it falls over the wedge shape. Through the use of the NURBS smoothing algorithm, the sheet is given the appearance of curving over the leading top edge of the wedge (prints B and C). Once settled (print D), the thermal softening system is activated, causing the stiffness of the springs to weaken thus causing elongation. In the wedge simulation this results in the curve of the sheet flattening as the control points fall closer to the floor. As demonstrated in section 5.2.4.3, when the sheet is heated upon the wedge the model should flatten along the leading edge of the polyhedron, causing the sheet to settle at a higher gradient along the ground, rather than the lower gradient that actually occurs.

Figure 5.24 – A Industrial Grade Sheet Melting Over the Triangular Wedge



Sheet Thermal Softening and Melting Using the Directional (Radiation) Virtual Heater System

Figure 5.25 illustrates the behaviour of a SaMM Sheet (with and without the NURBS system) as it falls over a wedge shaped polyhedra. As the sheet falls onto the shape (print A) it elongates (print B) before eventually settling (print C) along the centre of the wedge. As the effects of the thermal softening upon the sheet take effect, the right side of the sheet begins to melt and extend along the right side of the polyhedra while the left side of the shape remains static (prints D, E and F).

Figure 5.25 – Directional Virtual Heat on SaMM Sheet Deforming over a Wedge

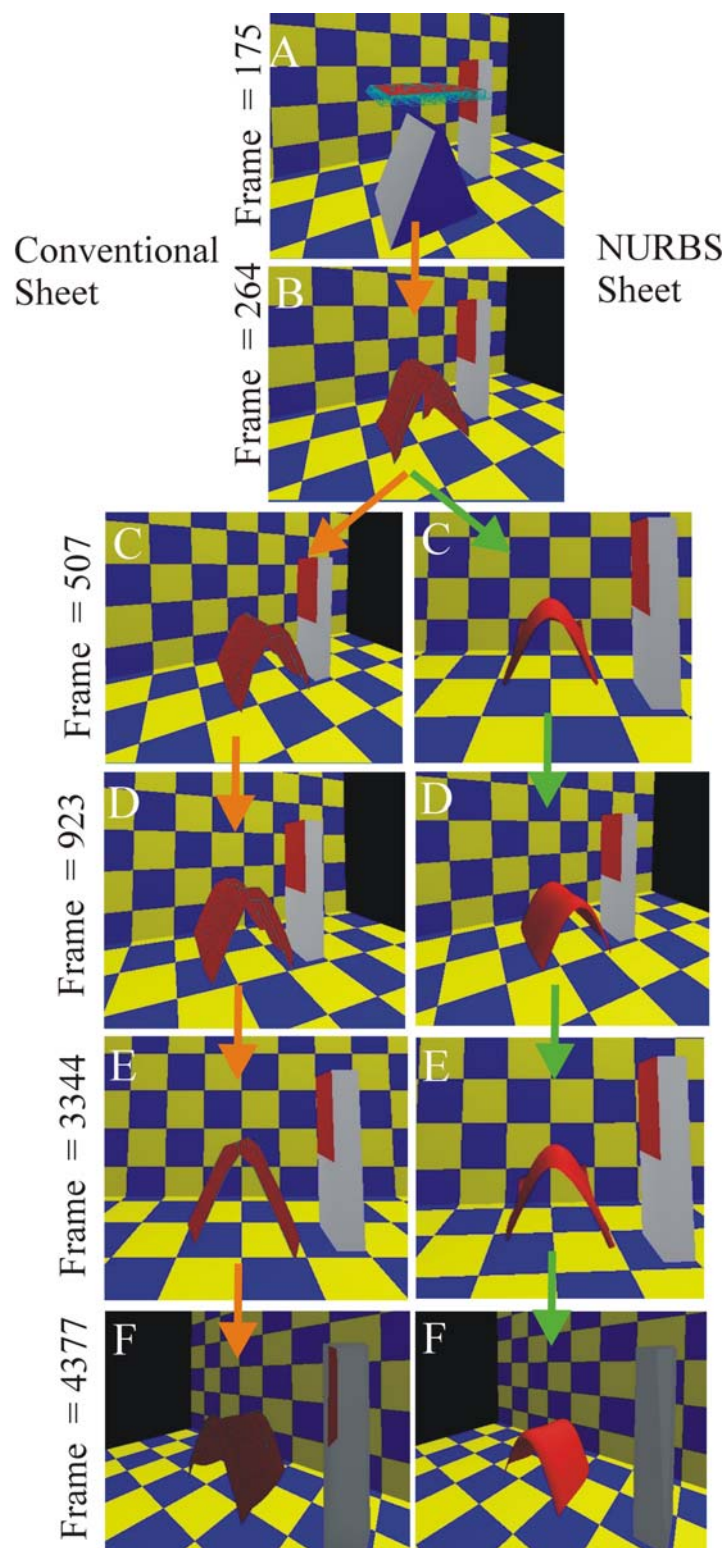
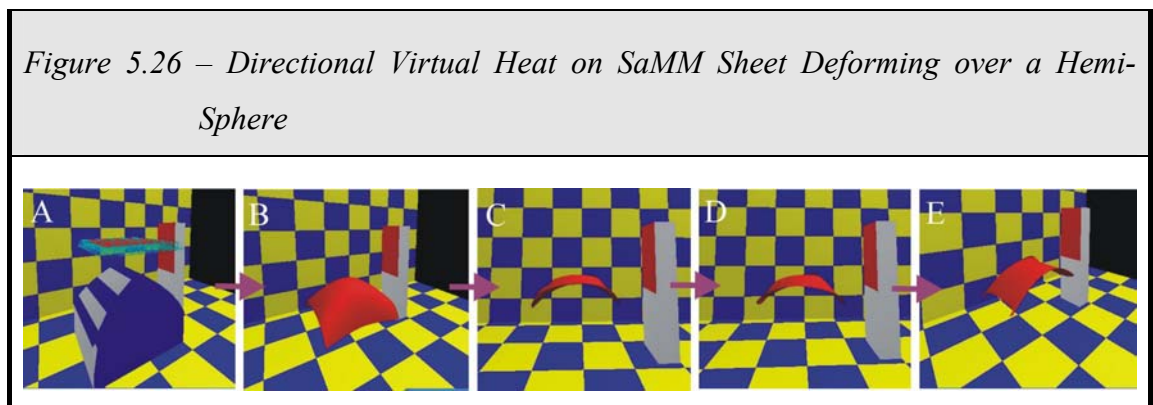


Figure 5.26 highlights a SaMM NURBS Sheet draping upon hemi-spherical polyhedra. As the sheet falls onto the shape (print A) it elongates (print B), before contracting and settling on top of the sphere (print C). As the effects of the thermal softening sheet take effect, there seems to be no clear indication as both sides of sheet melt and extend over the length of the polyhedra (prints C, D and E) in a similar manner to the oven thermal softening system.



Summary of Sheet Dynamics Testing

The extensive testing strategy employed here has highlighted the flexibility and stability that has been achieved through the use of three-dimensional sheet structures. Although the system was unable to achieve a high level of bending moment through the use of the layer sheet system alone, the employment of the NURBS-based system has overcome this shortfall producing a realistic smoothed surface. This approach has come at a price, resulting in accuracy issues with determining collisions when employing the NURBS system although it is felt that this inaccuracy could be minimised through the employment of a dynamic NURBS that adapts itself based upon the positioning of its control points. Alternatively, this shortcoming may be eliminated completely by integrating a refined NURBS system with the collision detection algorithm.

The thermal softening based tests have illustrated that the physical characteristics of a sheet can be adapted to produce a realistic interpretation of sheet melting, although this is dependent upon the precise nature of the simulation as rounded or smoothed surfaces provide substantially enhanced results over blunt-edged counterparts. Integration of the NURBS system with the collision detection system would once again significantly enhance the visual scope of the system over a wider assortment of simulations. The ability of the dynamics engine to adapt its proportions to maintain a comparable sheet surface area, only aids the realism that the system provides.

Through the real world tests, a number of characteristics with the melting algorithm were identified but not considered during the simulation behaviour. These included:

- The melting and rippling effect upon the sheet as the plastic liquefies and solidifies on the heated polyhedron.
- The generation of holes within the plastic as the build up of virtual heat within the polyhedron results in hot spots in the covered polyhedron forcing heated air to concentrate in specific areas, thereby causing the plastic to ripple, coalesce and eventually lead to the formation of holes.
- These issues represent specific material characteristics that vary dependent upon the melting characteristics of the plastic and the direction and type of heat being applied. For any real-time, visually aesthetic simulation the bubbling and rippling effects would be most efficiently handled through shading and texture mapping of the surface rather than actual physical modelling upon a surface itself. This approach would be significantly faster, numerically efficient and visually accurate; hence its widespread use in computer games. The generation of holes during the heating process could be resolved through the use of a remeshing algorithm which is examined in greater depth in Chapter 8.

The results for the simulated directional heat testing have not proven to be as effective as the oven thermal softening system. The approach was expected to produce significant thermal softening based deformation results with enhanced realism due to its focused approach to heat simulation. However a number of issues were encountered over the simple oven thermal softening system. These included:

- The oven thermal softening system has the capability of being switched on at any point during the simulation with the amount of virtual heat being exposed to the sheet also capable of being varied. This is not the case with the virtual heater system with the plastic sensitivity and thermal conductivity values of the sheet being used to adapt the level of deformation encountered by the sheet. However this still resulted in the whole sheet receiving a large dose of virtual heat prior to it colliding with the polyhedra, this minimising its visual effectiveness over the indiscriminate system.
- The thermal conductivity system to provide sufficient levels of conductivity within the sheet while maintaining system stability. This combined with its additional computational overhead ultimately reduced the effectiveness and flexibility of the system over the oven thermal softening method.
- Although the directional virtual heat system did achieve correct results for the wedge-based simulation (see figure 5.25), it was unable to produce results that were sufficiently distinguishable from the oven thermal softening system for the hemi-spherical experiment (see figure 5.26).

5.3.5 Results of MaSSE Engine Performance Testing

During the course of experimental testing, a number of idiosyncratic and interesting traits among the calculation engines were uncovered. These prevented a direct and balanced comparison between the differential equation approaches being made. The experimental work highlighted how, for a given scene, each MaSSE engine required significantly modified sheet and simulation parameters in order to gain a realistic and stable animation.

These results indicated that:

- The Runge-Kutta fourth order method proves a stable and realistic model with high levels of spring stiffness to retain the shape of the sheet under periods of high load or stress, with low levels of damping thus preventing the sheet from behaving in a anaesthetised or static style. This was not unexpected as the

fourth-order Runge-Kutta approach has long since been considered the benchmark for most applications requiring resolution through use of differential equations. The downside of this accuracy is the additional computational overhead that accompanies the technique, although this did prove to be less significant than expected with simulation performance (on average) only resulting in a 15-20% increase in real-time calculation time over the second order method. The drawback of this and the second order method were that they did not cater sufficiently for weak-spring sheets as the additional movement of the springs during a collision, combined with their low levels of damping often resulted in the sheet becoming unstable.

- The second order (Euler-Richardson) approach has proved to be the favoured means of all the techniques through its flexible nature in terms of high spring stiffness, low damping and stable nature without the high levels of computational overhead associated with the fourth-order approach. Once again the approach did not adapt well to weak-spring sheet systems although this was not considered to be a major drawback.
- The first-order (or Euler-Cromer) calculation required significant levels of damping with weak spring stiffness values to prevent a build up of energy into the sheet system. If not prevented, the sheet will eventually become unstable and explode. Attempts to find a suitable combination of spring stiffness, mass value and damping settings proved troublesome with anything other than accurate simulation parameters often resulting in oscillating sheets or completely unstable structures if not suitably contained (through the use of a reduced time-step).
- The Analytical engine behaves in a similar manner to the first-order calculation with the noticeable setback of being even more sensitive to simulation conditions. This too requires weak values for spring stiffness and even higher levels of damping to extract energy from the system when interacting with external forces. Perhaps its greatest drawback though is the need for even smaller time-steps which significantly undermine any benefits over the Euler-Cromer approach.

Experimental work has focused upon three different areas, analysing the performance of the three principal MaSSE approaches. Due to its instability and difficulty in achieving a realistic simulation, the analytical engine was not tested in any further detail.

Sheet Size Performance

The first test evaluated the significant performance overheads that were incurred upon the system through the use of varying sheet sizes. In these tests, the second-order Euler-Richardson system MaSSE engine was used on four TRiMM sheets as they were dropped onto two polyhedra. Although each sheet encompassed significantly different numbers of masses and springs (see Table 5.2) the sheet volume remained constant with the granularity of each sheet increasing with its mass-spring dimensions.

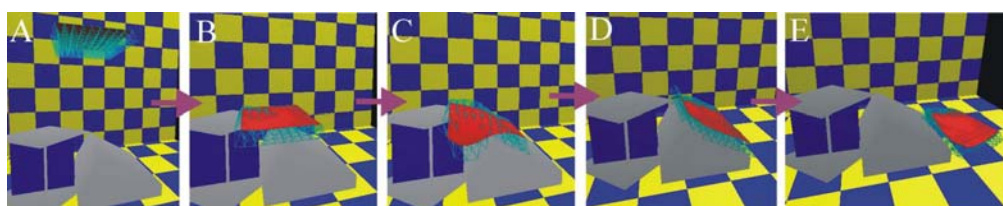
<i>Table 5.2 - Sheet Dimensions</i>		
Sheet Dimension (X- x Y- x Z-axis)	No of Masses	No of Springs
8 x 2 x 8	113	428
11 x 2 x 11	221	881
14 x 2 x 14	366	1496
17 x2 x 17	546	2273

Figure 5.27 illustrates a typical example of the scenario being measured. The testing examined the length of time it took for each sheet to fall under the effects of gravity from a predetermined distance above a single or group of polyhedra. The time measurements relate to the time from when the simulation is initiated to the point at which:

- The sheet collides with the *first* polyhedron (see print B).

- Collision occurs with the *second* polyhedron (see print C).
- The sheet slides down the length of the *second* polyhedron (see print D).
- Having slid off the polyhedron, the moment the sheet is in full contact with the environment floor (see print E).

Figure 5.27 - Sheet Size testing Scenario



Parameters for the simulation are detailed in Table 5.3 below.

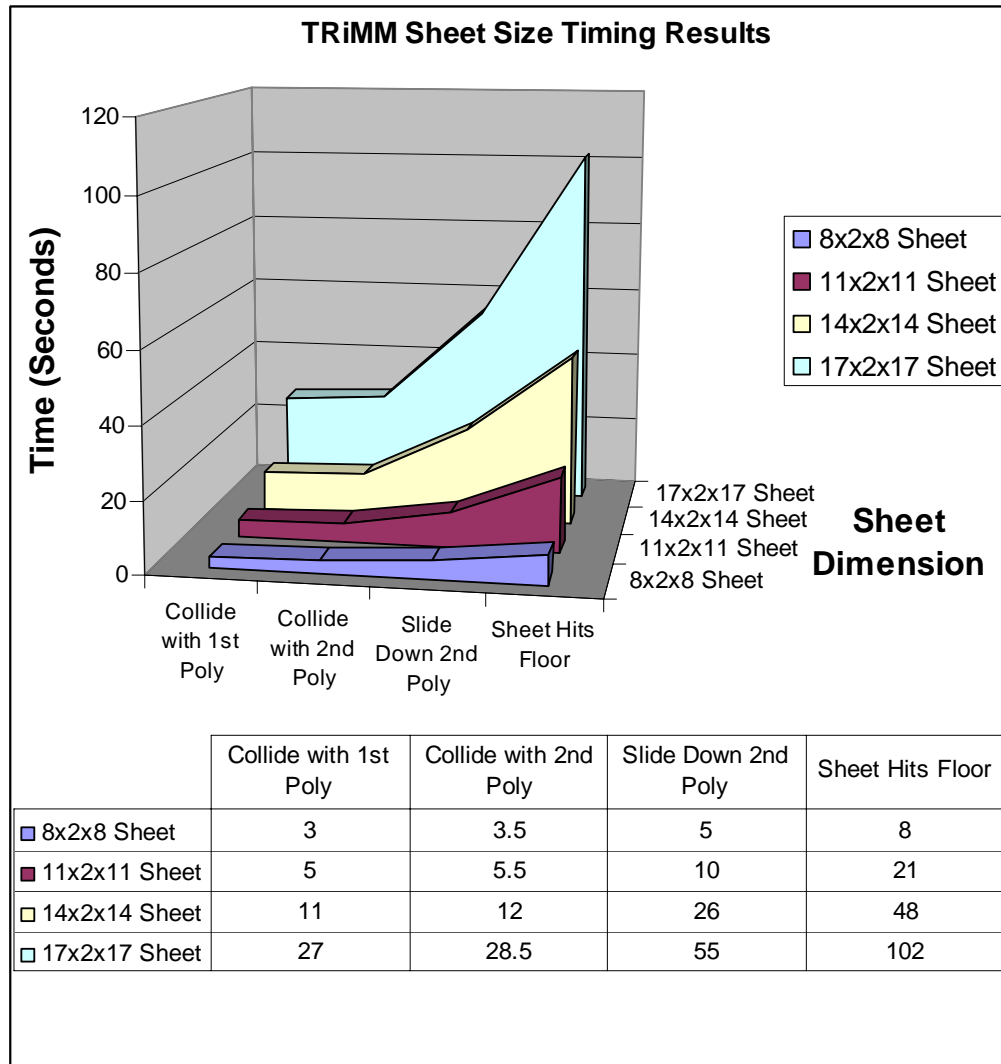
<i>Table 5.3 - MaSSE Engine Testing Variables</i>	
VARIABLE	VALUE
Spring stiffness (planar springs)	185 N/M x 10 ⁻³
Spring stiffness (inter-layer springs)	185 N/M x 10 ⁻³
Mass value for each Mass	0.4 KG x 10 ⁻⁴
Sheet Size (location in room) – X dimension (min-max)	-1.4 – 1.1 (2.5 x 10 ⁻² Metres)
Sheet Size (location in room) – Y dimension (min-max)	0.25 – 0.35 (0.1 x 10 ⁻² Metres)
Sheet Size (location in room) – Z dimension (min-max)	-1.4 – 1.1 (2.5 x 10 ⁻² Metres)

Room Size – X dimension (min-max)	-6.5 – 6.5 (13×10^{-2} Metres)
Room Size – Y dimension (min-max)	-2.81 (floor) – 3.0 (5.81×10^{-2} Metres)
Room Size – Z dimension (min-max)	-2.99 (back wall) – 3.33 (6.32×10^{-2} Metres)
Air damping coefficient	0.97 (3% coefficient)
Sheet damping coefficient	11.5% coefficient
MaSSE Engine	2 nd Order Integration
Time-step	0.015 Seconds

As the time results displayed in Figure 5.28 clearly identify, there is an exponential increase in the processing required by the simulation as the sheet dimensions are augmented. This is unsurprising given the increased numbers of masses and springs used as the dimensions are enlarged (see Table 5.2). The results also highlight the flexibility of the 11x2x11 sheet sizes to produce a workable simulation while providing effective real-time performance.

Given the modest performance of the test hardware and the significant scope for optimisation of the existing system (the tests were conducted in test/debug mode) there is significant potential to increase the sheet size to the larger dimensions with only a minor penalty in performance.

Figure 5.28 - Simulation Result Timings for 4 Varying Sheet Sizes

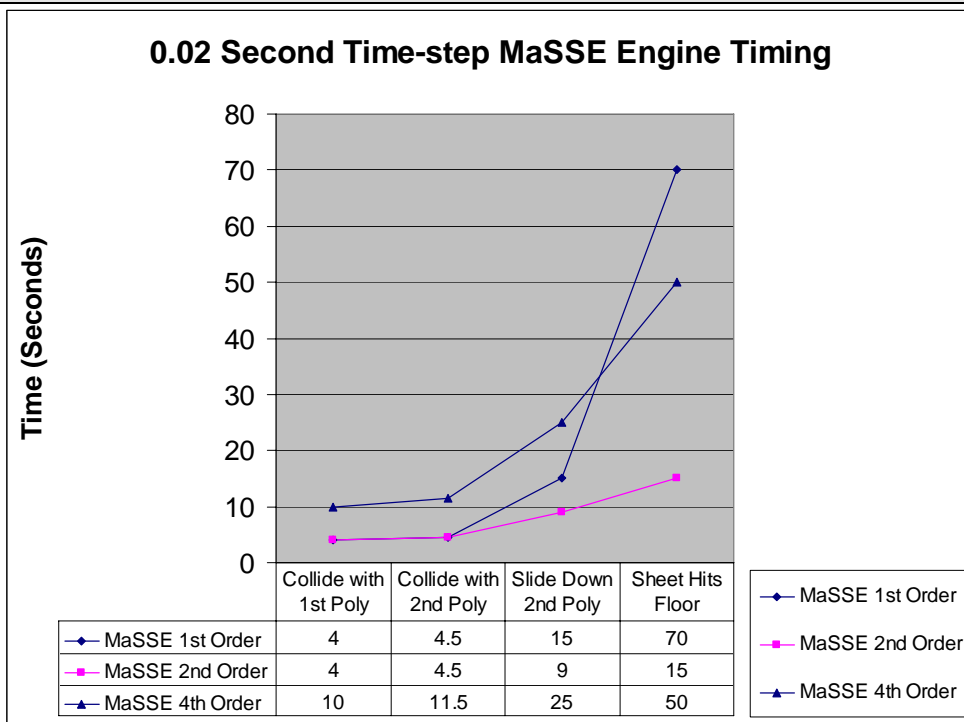


MaSSE Engine Performance Evaluation

The second set of testing is based upon the scene described in section 5.3.5 measuring the time performance of the 3 MaSSE engines when simulating an 11x2x11 TriMM sheet as it collided with the polyhedra. The time-step for all three scenes was set at 0.20 seconds to determine the overall time required to complete a simple scenario.

The results of these tests are detailed in Figure 5.29, with identical parameters (with the aforementioned exception of time-step) to those detailed in Table 5.3.

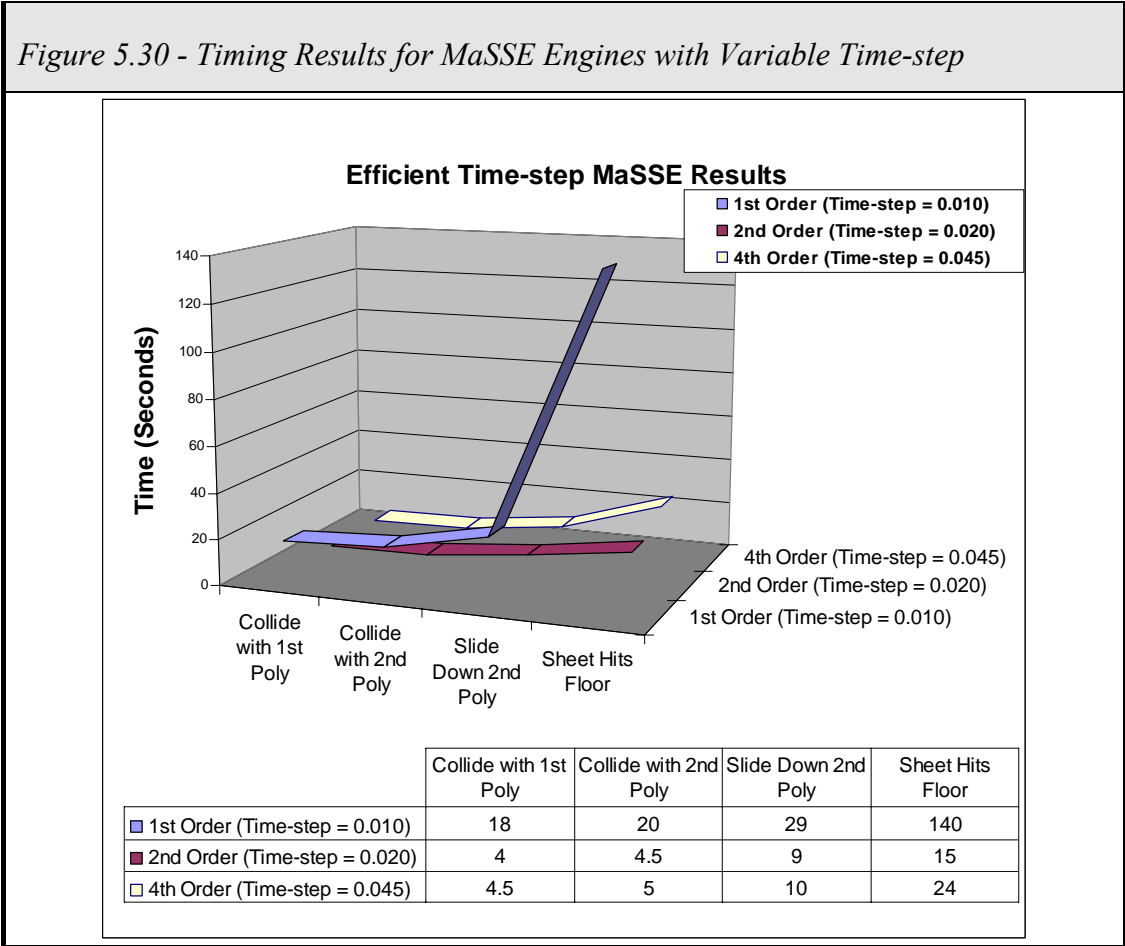
Figure 5.29 - Simulation Timing Results for MaSSE Engines with Fixed Time-step



At first glance, the first-order results appear surprising, with the sheet taking longer than the second-order sheet to start sliding down the polyhedron and the greatest time to slide onto the floor. However, the results fail to produce the true picture of events as (due to inaccuracies with using first-order calculations combined with the use of such a large time-step) the sheet appeared to become entangled over both polyhedrons and when it eventually started to slide down the sloping face of the polyhedron, did so at a substantially slower rate than the second and fourth-order sheets. The results from the second and fourth-order simulations provided more predictable behaviour with the fourth-order Runge-Kutta simulation taking more than twice as long to complete its simulation than the second-order Euler-Richardson system. Some of this additional time over the second-order approach is attributable to the simulation PC running low of physical memory towards the end of the simulation, resulting in a further reduction in performance as the PC was forced to use slower, paged, virtual memory.

Performance of MaSSE Engines with Adapted Time-Step

As the choice of MaSSE engine has a significant influence upon several simulation parameters, most notably time-step, the third group of tests duplicate the experiments set out in section 5.3.5, but with time-step values for each integration level that provide the most efficient, stable result whilst producing a realistic simulation of sheet behaviour. Following a series of preliminary trials with varying time-step values for each integration engine to determine their greatest step size, a set of timing results were produced demonstrating the speed of each optimised integration method to complete the simulation scene described in section 5.3.5. These results are presented in Figure 5.30.



Once again the inaccuracy of the first-order approach proved to produce non-representative results as the velocity of the sheet was significantly smaller than those of the second and fourth-order sheets, resulting in increased times to start sliding and to reach the floor of the environment.

Once given the opportunity to extend itself, the robustness of the fourth-order Runge-Kutta method allowed for a substantially larger step-size than the Euler-Richardson approach (0.045 seconds compared to 0.020 seconds) resulting in a significant improvement in its real-time efficiency being less than 1 second slower than the second-order approach through the first three stages. The additional benefits of the Runge-Kutta approach over the Euler-Cromer (i.e. greater stability, smoother sheet visual performance) were also evident in the tests. However, so too was its Achilles Heel, the fact that substantial levels of resources are required to produce sufficient performance. Although in modern desktop systems the use of the method would be minimised, under the test conditions the intense calculations and memory demands would monopolise the system for the course of the simulation.

5.4 Summary of Results

The results of this experimental chapter clearly demonstrate that a three-dimensional matrix structure for mass-spring systems does indeed provide a workable solution to sheet modelling. Although the approach has not produced the high levels of inter-plane stiffness across the sheet as expected, the use of a NURBS system has minimised this shortfall by producing smoothed surfaces across the sheet surface. The virtual heat algorithms have proven effective at mimicking thermal softening across a surface although testing has shown that the results are dependent upon the nature of the scene being modelled.

The MaSSE engine analysis has highlighted a number of differences between the three approaches. The Euler-Cromer first-order approach highlighted the inefficiencies of the method at producing accurate simulations. The employment of complex mass-spring networks requires significant levels of accuracy to ensure numerical stability is maintained. Testing has highlighted that, even once a suitable medium of parameters can be identified; producing a life-like solution is troublesome. Common issues include harmonic vibration of the sheet or in an attempt to eliminate this, system over-damping. Even minor changes in simulation parameters (such as variation in the polyhedron shape, or the sheet drop distance) could result in the scene failing to produce a stable or life-like solution. To demonstrate the relationship between these parameters, a change to increase the initial height of a falling sheet onto a polyhedron would require enhanced sheet damping (to maintain energy levels within the sheet during the collision), larger levels of spring stiffness (to retain sheet structure/shape), reduced mass-values and reduced time-step intervals.

The Euler-Richardson method is significantly more robust catering for a wider variation of parameters, enhanced stability, visual accuracy and increased time-step without sacrificing performance. This has proved to be the chosen method for sheet simulation.

The Runge-Kutta approach has produced enhanced results and stability over the second-order system with the added benefit of ruggedness that allows substantial adaptation of the sheet scene without having to adapt the simulation parameters. All of this does come at the expense of performance with the technique losing out to the second-order system due to this set-back. However, the surprising pace of the fourth-order system

when compared to the second-order system when using efficient time-steps would indicate that the Runge-Kutta method is the most suitable method when sufficient computing resources are available.

Table 5.4 details a typical set of parameters for the three MaSSE engines with the Runge-Kutta values allowing for a significant adaptation from these values ($\pm 100\%$), with Euler-Richardson catering for reduced levels of change (40%) with the first-order Euler-Cromer system allowing minimal variation.

<i>Table 5.4 - Typical sheet parameters for MaSSE Engines</i>			
MaSSE Engine	Spring Stiffness (N/M $\times 10^{-3}$)	Sheet Damping (% Coefficient)	Adequate Time-step (Seconds)
First Order	60	34	0.010
Second Order	200	14	0.020
Fourth Order	400	14	0.040

6 Human Computer Interaction Experimentation

6.1 Introduction

The basis of this chapter is draw upon Human Computer Interaction (HCI) principles to prove or disprove the hypothesis component that requires that the resulting methodology should produce an aesthetically accurate model for sheet deformation.

A secondary element of the study has evaluated the hypothesis declaration that the system is ‘computationally efficient’ by drawing comparisons between the relative speed performances of the sheets in otherwise identical simulations.

To evaluate this requirement, HCI testing has been employed to determine the realistic effectiveness of the three-dimensional mass-spring approach to the real world experimentation that are covered in Chapter 5, but to contrast this new methodology to an existing two-dimensional mass-spring approach. This chapter represents the results of a preliminary review and was not intended to form an exhaustive evaluation on the aesthetic benefits of different sheet structures.

6.2 The System Bench Mark

As discussed in chapter 3, the “flexion” system reported by Provot (1995) has shown to be an effective cloth simulation approach through the use of two-dimensional lattice structures. It is for this reason that the flexion system is to be the benchmark against which the MaSSE system will be measured.

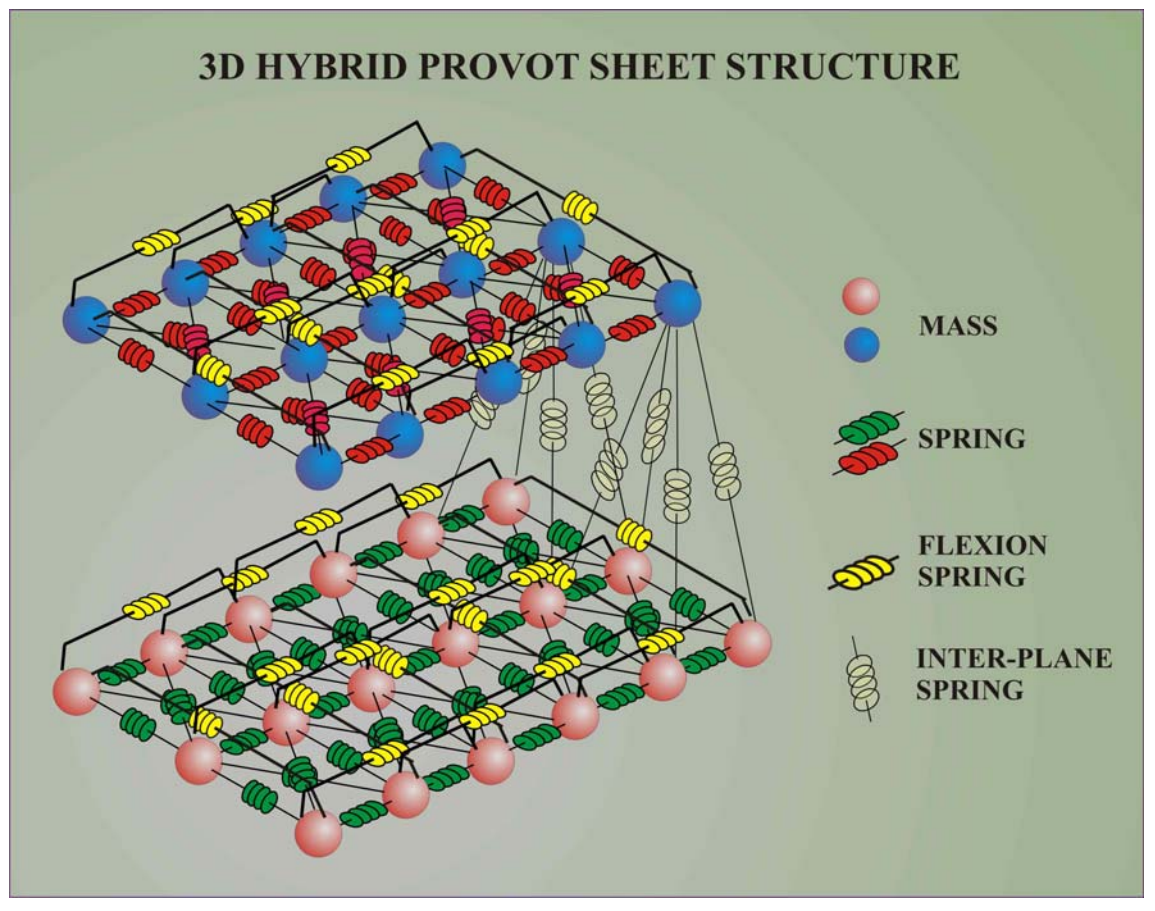
It is not possible to get the source-code for the flexion system, but instead the MaSSE system was extended using the published description to incorporate Provot’s flexion springs onto a two-dimensional MaSSE sheet. Care was taken to ensure that all other sheet parameters, such as spring stiffness and internal damping remained constant whilst maintaining simulation stability.

The opportunity was also taken to adopt an additional level of testing to evaluate the performance of a three-dimensional implementation of the Provot structure using a two tier matrix such as those used in the SaMM and SLaMM models. The performance of this hybrid system was then tested alongside the SLaMM and TriMM models in the HCI appraisal.

A pictorial representation of the hybrid 3D Provot model is illustrated in Figure 6.1.

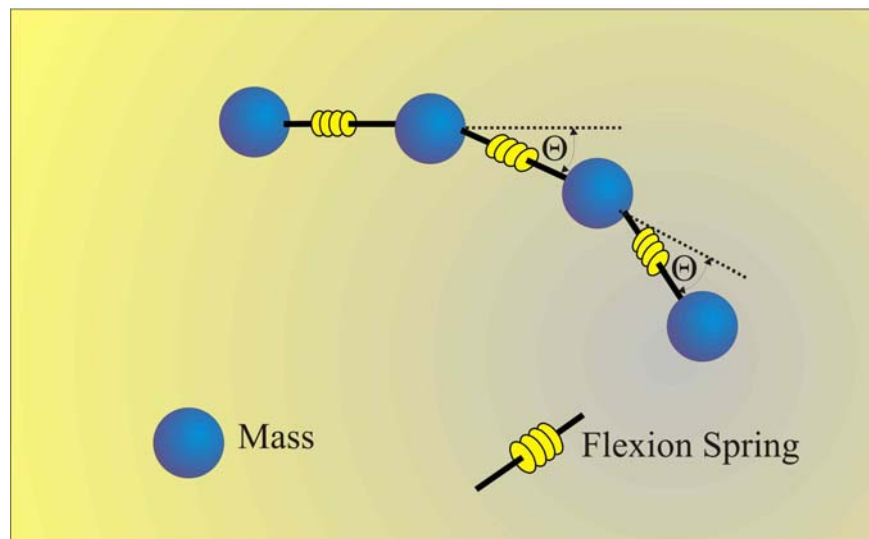
The hybrid system incorporates the use of flexion ‘rotational stiffness’ springs with a 2-layer SLAMM matrix that also integrates a 2nd diagonal inter-plane cross-member. Conventional inter-plane springs are used to link masses between the two planes as is the case with the conventional SaMM, SLaMM and TRiMM structures. The flexion springs connect second-neighbouring masses (i.e. the masses that are direct neighbours to its immediate neighbours in a given direction).

Figure 6.1 – Structure of the 3D Hybrid Provot Sheet



The flexion springs adopt an established approach to modelling rotational stiffness through calculation of the normal plane between the every flexion connected groups of masses before computing whether the angular difference between any two planes that link to a given mass exceed a predetermined tolerance (indicated by θ in figure 6.2). if this angle is exceeded then restrictions are placed on the angle of the second plane to ensure that θ remains at the preset rotational value.

Figure 6.2 – Flexion Rotational Stiffness Model



6.3 HCI Experimental Testing

To prove or disprove the hypothesis that the use of three-dimensional sheet structures provides an aesthetically accurate representation of sheet deformation, experimental testing was conducted to gauge opinion on the realism of the sheet system compared with real world video. To assess the effectiveness of three-dimensional sheet structures, the experimentation measured the realism of using two-dimensional synthetic sheet structures to those of their real world and synthetic three-dimensional counterparts. It was felt that the SaMM provided no additional benefits over the SLaMM model and so was excluded from the HCI tests.

In addition to the two advanced three-dimensional sheet structures, SLaMM and TriMM, and the benchmark two-dimensional Provot system, the opportunity was taken to form a hybrid three-dimensional sheet model that encompassed the flexion structure within a three-dimensional sheet model.

6.3.1 The Evaluation Test

The test procedure involved each test subject completing two questionnaires, the first was a Background Questionnaire asking the test subject to provide basic information on their knowledge of engineering and IT principles. This was then followed by the multiple-choice User Evaluation Questionnaire.

The candidates were first shown the real-world video footage of the sheet falling onto a Polyedron. The purpose of this exercise was to provide the user with a visual benchmark with which to base their scoring of the synthetic representation.

The subjects were then shown the applicable simulation using either of the four sheet models before then being shown asked to complete Part 1 of the User Evaluation Questionnaire.

The remaining three sheet models were then demonstrated with the test subject being asked to complete Part 1 of the User Evaluation Questionnaire immediately after each simulation.

The process of viewing the real-world footage, viewing the 4 respective sheet simulations while marking the performance of each sheet model on Part 1 of the User Evaluation Form was then repeated for the remaining 2 experiments.

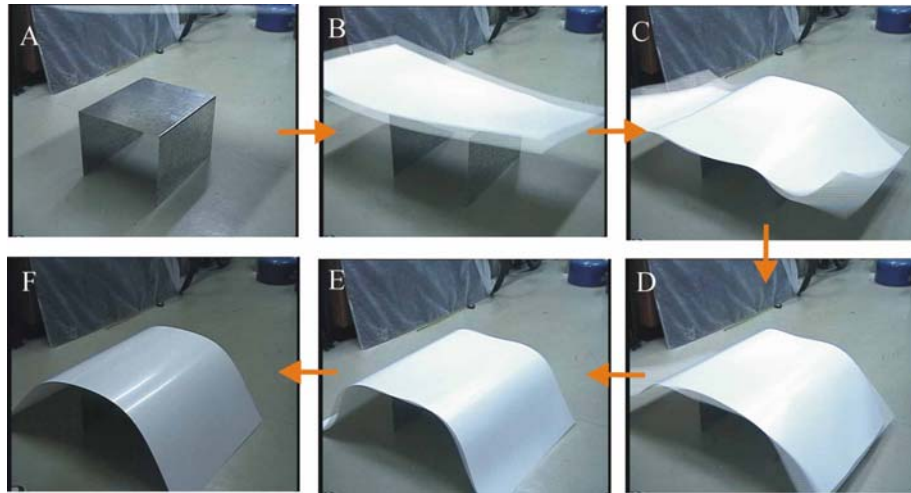
At the end of the evaluation, the test subjects were then asked to complete Part 2 of the User Evaluation Questionnaire.

Example test questionnaires are appended to Appendix D.

The basis of the testing process was to independently rate the visual performance and speed of each sheet model as they undergo three very different types of simulation. To ensure that the test subject's centre of attention remained focused upon the sheet, the Polyhedron was set to 'hidden' prior to the collision occurring, as were all of the sheet's masses and springs. The NURBS functionality was activated during all of the simulations.

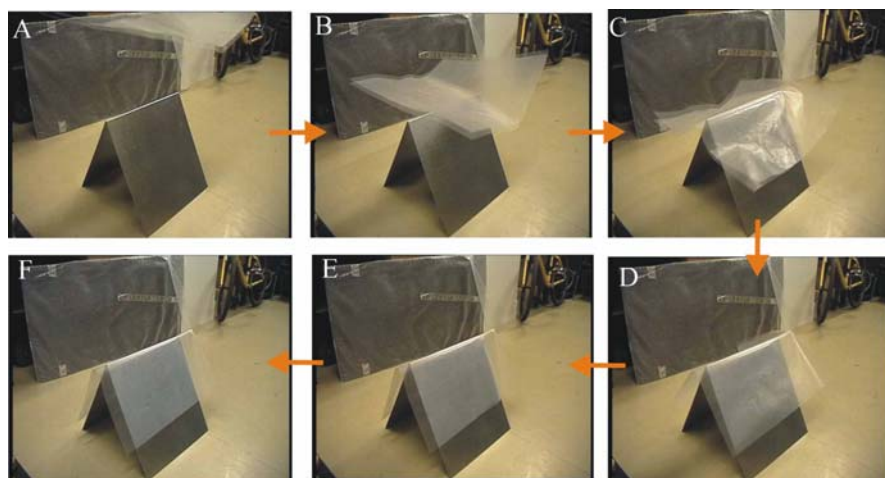
The first evaluation focused upon the dropping a sheet onto a cube. The real-world footage is presented in figure 6.2. The purpose of this simulation was to demonstrate the sheet dropping onto an awkward shaped polyhedron since each sheet will extend over the edge of the cube's flat top surface. For this experiment, the 4th order Runge-Kutta MaSSE engine was used. The sheet springs were all set to a medium degree of stiffness.

Figure 6.2 – First Evaluation: Styrene Sheet Falling onto a Cube



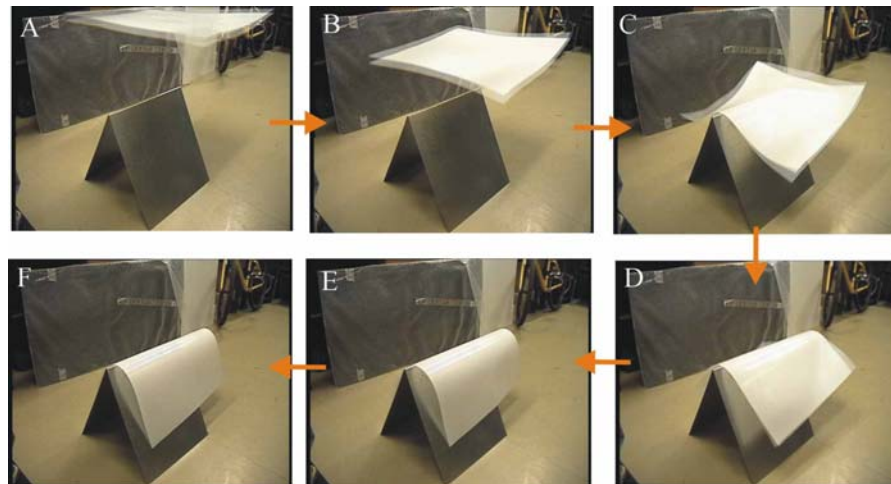
The next evaluation concentrated upon studying the behaviour of dropping a highly elastic sheet onto the wedge polyhedron, as presented in the real-world footage in figure 6.3. The role of this simulation was to demonstrate the how the material properties of the sheets could be varied to exercise this highly elastic behaviour and how each sheet structure coped with these settings. Once again the 4th order Runge-Kutta MaSSE engine was used with the sheet springs being set to a high stiffness value. Visually this is not a complex simulation and as such will not push the sheet's to their full potential, but will outline their suitability for modelling highly elastic materials.

Figure 6.3 – Second Evaluation: Industrial Grade Plastic Sheet Falling onto a Wedge



The third study was used demonstrate the highly dynamic capabilities of each sheet and to access the suitability of using only the 2nd order MaSSE engine upon the behaviour of each sheet structure. The real-world video footage, as presented in figure 6.4, demonstrated a styrene sheet falling and resting onto the centre of the wedge but the users were each informed that the simulation they will see shall demonstrate more dynamic behaviour. The 2nd order Euler-Richardson MaSSE engine was used in conjunction with weak levels of spring stiffness within the sheet. Of the three tests, this experiment will demonstrate most clearly how each model copes with vibrant flexible structures with a fast integration engine.

Figure 6.4 – Third Evaluation: Styrene Sheet Falling onto a Wedge



6.3.2 The Questionnaires

The underlying principle behind the questionnaire design was to establish if the test population felt that the three-dimensional sheet structures provided a greater degree of realism for modelling sheet behaviour than the traditional two-dimensional structure developed by Provot. A second element of the testing was to establish if the users felt that the multi-layered models provided sufficient performance in terms of computational speed relative to their single layer counterparts.

A number of parameters were used in the evaluation to measure the level of realism the sheet models provide. These include:

- a) The initial post-collision deformation of the sheet.
- b) The bending, stretching and/or sliding behaviour.
- c) Reforming and settled shape retention performance.
- d) The model's speed performance while undergoing elastic deformation.
- e) Overall realism of the system.

Once the data has been gathered, statistical analysis was performed to analyse user's opinion of the system with respect to its effectiveness in achieving the hypothesis aims.

6.3.3 The Test Population

As the testing requirement for these experiments comprise of basic computing skills and good visual attentiveness, these were used as the basis for selection criteria for the test candidates. A suitable population size was devised based upon an effectiveness number of tests to prove or disprove the hypothesis.

In total, a test population of 10 individuals were used. This comprised of 8 men and 2 women with the age group varying from 23 to 45. All participants were educated and covered a breadth of engineering and non-engineering based professions. All the subjects had a good grasp of computing skills with an average of 9.9 years computing experience and the entire population typically spent over 20 hours a week using a computer.

Only 30% of the population were familiar with computer based simulation (excluding games) while 50% were knowledgeable on advanced Engineering and/or Physics based principles.

6.4 HCI Test Results

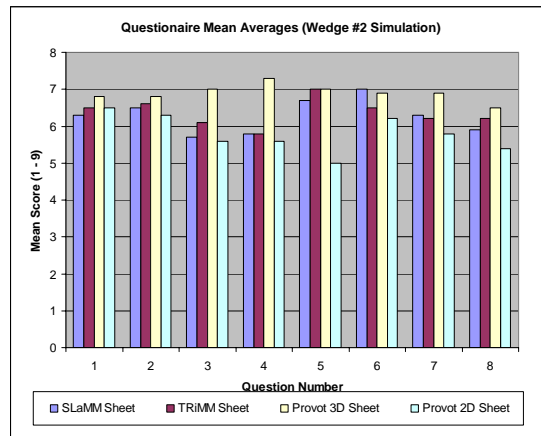
The unprocessed test results are presented in Appendix D.

Questions 1 to 5 relate to the deformation performance of the sheet and questions 6 to 8 examine the speed performance of the simulation.

Figures 6.5a, 6.5b and 6.5c represent the mean value scores for the 3 tests. In the cube based tests, figure 6.5a signifies that the TriMM structure offers superior performance over the 2D Provot system, while in the wedge#1 simulation, the Provot 2D offered the best visual and speed performance, with the SLaMM model coming a close second in most areas other than sliding performance where it showed a clear lead. During the second wedge#2 experiment, the 3D Provot sheet demonstrated a clear advantage over all of the other sheet models.



Figure 6.5c –Wedge#2 Mean Values



The mean value results are validated when the median figures for the experiments are examined in figures 6.6a, 6.6b and 6.6c. Again the 3D hybrid Provot system produced splendid results in the second wedge results, with a strong showing in the first wedge test. The TriMM model performed well in the majority of the cube test ratings, only succumbing to the two Provot-based models in the speed performance tests. This continued in the second wedge test where it, along with the SaMM structure outperformed the 2D Provot system, only to be beaten by the 3D Provot model. However the TriMM results weren't maintained in the first wedge test where the model lagged behind the other 3 models.

Figure 6.6a –Cube Median Values

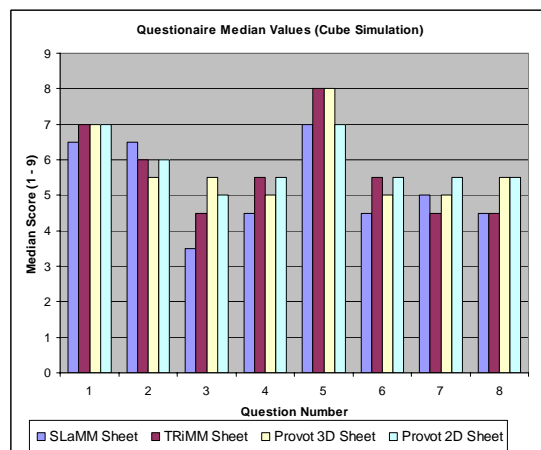


Figure 6.6b – Wedge#1 Median Values

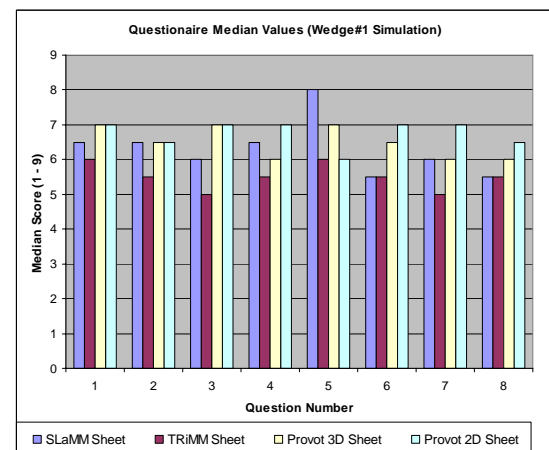
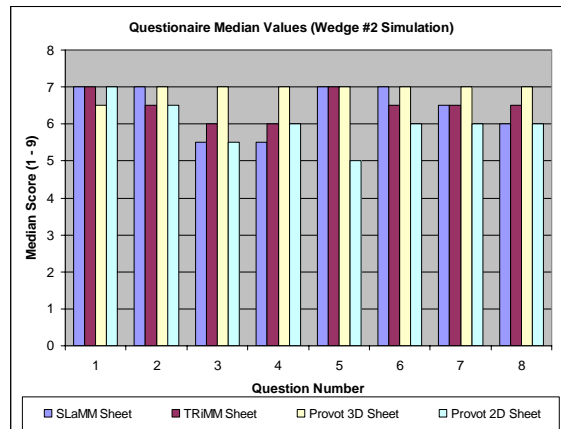


Figure 6.6c - Wedge#2 Median Values



The standard deviation values demonstrate the low level of divergence for the 3D Provot system, particularly in the wedge#2 tests where it produced consistent results. In the same test, the 2D Provot deviations also remained low compared to the TriMM and SLaMM models. Once again, the Provot 3D variations remained low for the visual performance questions in the cube and wedge#1 test, allowing it to outperform the other models.

Figure 6.7a – Cube Standard Dev.

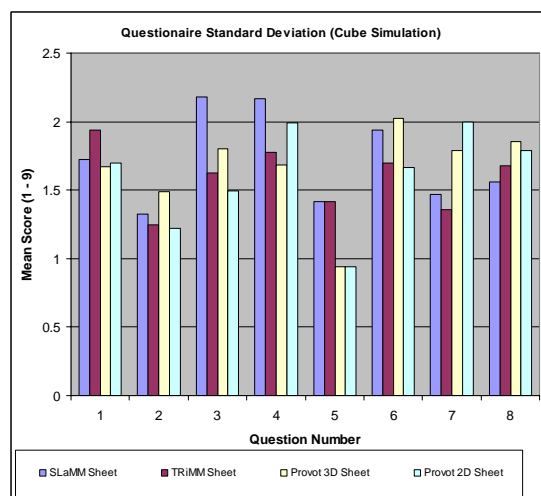


Figure 6.7b – Wedge#1 Standard Dev.

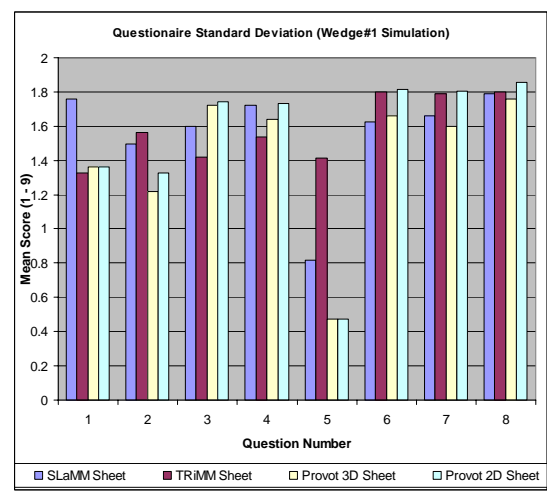
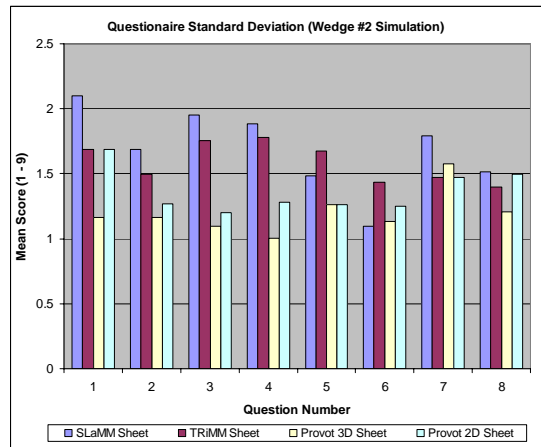


Figure 6.7c – Wedge#2 Standard Dev.



The overall mean results (see figures 6.8a and 6.8b) for question C1 highlight a positive response (with a score of 6.0) concerning the accuracy of the simulations and how they relate to the real-world video footage, with a marginally lower scoring of 5.8 for question C2 reflecting the overall feeling of the sheet simulations as a whole. The deviation in scores was greater for question C2 with a standard deviation of 1.83, as opposed to C1's figure of 1.41.

Figure 6.8a – Overall Mean Results

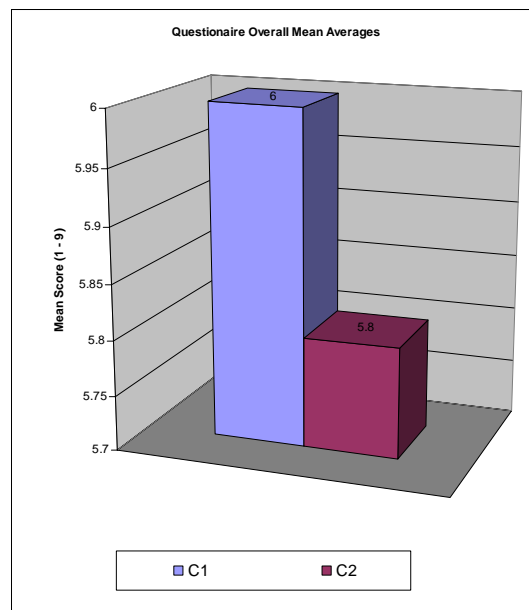
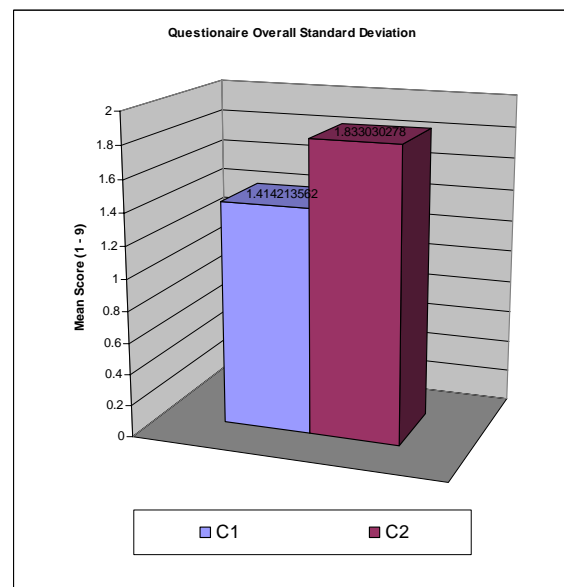


Figure 6.8b – Overall Standard Deviation



6.5 Summary of Test Results

The results indicate that the hybrid 3D Provot system performed better than both the 2D and other multilayer meshes, producing results that are certainly worthy of further investigation. Consistently scoring high marks in all three tests, but most importantly in the third Euler-Richardson 2nd order experiment, where it came out as the clear overall winner in both the mean and median scores.

The 2D Provot system failed to match the overall results of the multi-layer systems, in particular the TriMM and the 3D Provot system, often finding it trading scores with the SLaMM sheet. Where it excelled was in the (least important) wedge#1 simulation where the sheets, instilled with a very low level of rotational stiffness, played to its strengths as a fast and easily flexible sheet structure. Here the added complexity of the multi-layer meshes often resulted in slightly slower performance and reduced realism as the multi-layer meshes still produced an unwarranted bending moment. This issue could be easily overcome through some careful manipulation of the interlayer spring stiffness values, although it is unlikely that the multi-layer models would match the performance of a 2D system.

The TriMM structure performed well across the board, producing marginally better marks than the SaMM and 2D Provot systems although failing to match the scores produced by the 3D Provot mesh. In the all-important wedge#2 test, the model gave an admiral performance, resulting in a positive ranking over the 2D Provot mesh.

The two MaSSE meshes tended to have a wider variation in scoring results than the Provot Systems, particularly the SLaMM model. The 2D Provot model did produce lower variations in its scores, although even these were not low enough to upstage the standard deviation values for the 3D Provot system.

7 Discussion

7.1 Introduction

This body of work has attempted to resolve a number of problems that exist when modelling real world behaviour in a real-time synthetic environment. These issues include the use of stable, explicit integration techniques to model sheets with a high level of rotational stiffness, evaluating the performance of three-dimensional mesh structures in relation to well a well documented two-dimensional model, analysing the behaviour of the models with respect to first-, second and fourth- order integration techniques. All this work was conducted in the attempt to produce an acceptably accurate system of sheet modelling, with additional research being conducted into the suitability of the system for modelling heat-induced deformation.

The exact aim of this investigation has been to evaluate the performance of mass-spring systems in the real-time simulation of sheet materials as they deform as a result of collisions with foreign bodies.

The study has

- Evaluated the performance of three distinct sheet structure designs
- Analysed the benefits and disadvantages of utilising first, second and fourth order differential equations to model the sheet behaviour.
- Generated a World on Window interface to allow the system to accurately model and control the sheet behaviour.
- Investigated the benefits of incorporating novel ideas into the development of the system, such as the use of three-dimensional layered sheets and use of a NURBS system to enhance the visual accuracy of the system.
- Produced a preliminary HCI review of the system, compared alongside a benchmark two-dimensional model and a hybrid multi-layer matrix that encompasses elements of both the benchmark system and that of the multi-layer approach.

In addition this body of work has also produced and evaluated two methods of thermal softening/virtual heat simulation that attempts to reproduce the effects of background and directional heat within the environment.

7.2 Fulfilment of Aims

This thesis has reported on the construction of a computationally inexpensive environment that allows a model to be manipulated and easily deformed with a sufficient level of accuracy.

The WoW environment was created to cater for the development and evaluation of modelling sheet materials in real-time using first-, second- and fourth- order integration methods.

Bespoke three-dimensional structures and polyhedra may then be installed into the synthetic system at run-time to test the behaviour of the integration methods and those of the three-dimensional meshes.

In addition to the core prototype system, that was used to evaluate the research findings, a model-generation application was created due to the exponential complexity of the meshes. The sheet structures allow an object to be given a variety of initial properties.

Comparisons were then made between the behaviour of the novel three-dimensional meshes and those of (i) real-world testing and (ii) an existing two-dimensional sheet structure based upon Xavier Provot's documented works (Provot, 1995).

A limited study was conducted to establish the suitability of the three-dimensional model to mimicking the effects of virtual heat-based deformation.

A preliminary HCI based study was conducted to evaluate the visual performance of the sheet structures with a hybrid multi-layer Provot-based system yielding the best results. The TriMM sheet structure did outperform the 2D Provot mesh; however the most intriguing results were borne from the behaviour of the hybrid sheet model, with the positive results from the 3D Provot mesh warranting further investigation.

The sheet simulation environment did perform well, producing simulations with a satisfactory level of realism in real-time using modest computing facilities.

The narrow study of virtual heat-induced deformation did prove that the oven thermal softening method was more effective approach for modelling thermally sensitive deformation, producing enhanced, more lifelike results with reduced computation than

the virtual heater-based system. Although unexpected, it has demonstrated the use of complex algorithms may not always produce the most favourable results.

The results of the system described have proven positive with all indications showing that the underlying aim has been achieved.

7.3 Further System Expansion

Following experimental testing, it was felt that not all aspects of the system had achieved their full potential and may benefit from additional investigation to achieve the requirements laid out by the research hypothesis, the production of an aesthetically accurate representation of sheet bending and deformation in a real-time system.

Six aspects of the system have been identified as areas for further extension that shall now be individually examined in further detail.

- a) *The 3D Provot Mesh.* An investigation into the effectiveness of the 3D Provot system, particularly in relation to the TriMM model using second and fourth- order integration methods.
- b) *Active Sheet Manipulation in the WoW.* The WoW environment could be developed even further to allow full manipulation of the sheet structure (via its control points). From the perspective of Human Computer Interaction (HCI), this would instil the system with an additional level of interaction with the user not only to allow for a greater number of experimental simulations, but also enhance the simulation ‘experience’.
- c) *Optimisation of the MaSSE engine.* This would allow the system to cater for larger and more complex models while retaining its need for only modest computing facilities in its real-time performance. One avenue to pursue would be the use of dynamic re-meshing algorithms to minimise the system work-load (see areas for future research).
- d) *Possible completion/enhancement of the sheet Self-Collision Detection Algorithm.* To aid the realism of the sheet, a self collision system was installed that separated the sheet into four regular quadrants. Once a collision had taken place between a mass and any other object, a check was made between each mass and the faces of the mass’s three sibling sector’s to ensure that the mass has not passed through one of the face’s. Should this occur then the system will modify the behaviour of the

mass to reflect a collision with itself. The algorithm has been developed and installed although effects upon system performance, accuracy and potential benefits resulted in its deactivation from the final solution.

- e) *Adaptation of the NURBS system.* Two possible approaches would provide possible solutions. The first would be to adjust dynamically the smoothing algorithm to avoid penetration of the polyhedron surfaces. The second is to adapt a NURBS system that extrapolates and returns its interpretation of the mass control points along its smoothed surfaces which may then be fed directly into the collision detection system to adjust the positioning of the sheet as it contacts with the object.
- f) *Three-Dimensional Object Modelling.* Only laminate deformable structures have been tested with the system, although the facility does exist with the model-generation application to test three-dimensional deformable cubes and other bodies. Adapting both the generation and simulation system to cater for complex three-dimensional bodies would expand the capabilities and uses of the application.
- g) *Incorporation of GPU Programming.* As no GPU based programming was utilised in the development of the system, experimental work should be conducted into the performance benefits utilising the technology may have against the known limitations and restrictions the technique may bring to any eventual application.
- h) *Further Investigation of Verlet Integration.* Experimental work could be conducted into the suitability of employing Verlet integration methods compared to the Euler based methods currently employed. The performance and stability benefits will need to be evaluated against the restricted scope of simulations where the technique could be used.

7.4 Recommendations for Future Work

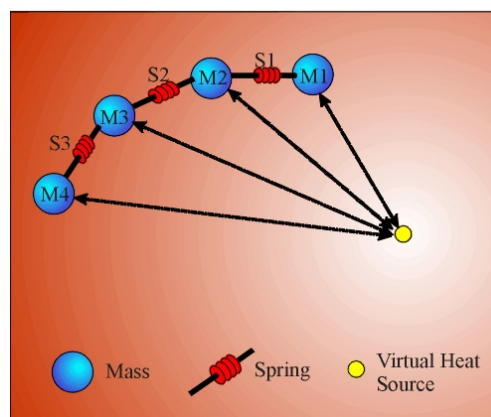
In answering the questions relating to the applicability of mass-spring systems in real-time modelling, this work has also raised a number of questions that go beyond the scope of this study. The following are suggestions for possible extensions to the existing MaSSE system/application that would answer these enquiries, which in turn may lead to further investigations.

- i. *Modelling of non-laminar models.* This would extend the capabilities of the system, allowing it to be evaluated for use with Free-Form Deformation algorithms. Although the sheet generation system does cater for the production of multiple level sheet structures, a more advanced system would overcome the height and shape limitations of the current system. Further experimentation with non-laminate structures would test the suitability of the MaSSE engine as a generic Mass-Spring computational engine.
- ii. *Implementation of a haptic force feedback device into the system.* As haptic interfaces become more prevalent in current research, adapting the WoW interface to cater for haptic interaction would not only widen the scope of use for the application, but its overall appeal.
- iii. *Experimental work with new sheet structures.* This would include developing an enhanced 3D Provot system, producing an adapted SLaMM structure, encompassing a second cross-member spring as well as refinement of an extended version of this new structure and the TRIMM model that contain additional cross-over springs similar to those in the 3D Provot system.
- iv. *Implementation of a Spring Retention Algorithm.* This could be used to help retain numerical stability within the sheet for simulation runs involving high energy levels with low damping settings. Initial work had already been completed but the decision was taken not to implement this safeguard-feature fully due to the lack of suitable benefits at that time.

- v. *The installation of a re-meshing algorithm.* This would augment the look and feel of the sheet as it sheet as it deforms, allowing tears to form along the laminate surface and cater for the formation of holes during virtual heat-based scenarios
- vi. *Advanced Implementation of a Virtual Heat System with dynamic hole-generation.* The facility to generate holes within the sheet as it tears or simply disintegrates (as in the real world heat experiments) would provide an extra level of realism to the simulation, which would particularly helpful in any future virtual heat simulation work. This could be achieved through the use of a re-meshing algorithm that could also be adapted to cater for the sporadic tearing of the sheet. Any system would work by automatically inserting or removing masses and springs to and from the sheet structure in areas where high levels of interaction are taking place.
- vii. Extending and experimentation of the Voigt-Kelvin element spring and dashpot model, which has been combined with a Maxwell element system to produce an effective model for simulating viscoelastic behaviour.
- viii. *Dynamic sheet-to-sheet sticking and crumpling.* This was an unexpected result of the real world experimental testing, which illustrated the varying effects that virtual heat has upon different plastic materials. This feature would be useful in any future advancement of the thermal softening system.
- ix. *Implementation with a fifth-order Runge-Kutta algorithm, incorporating error correction.* Although there is a substantial overhead with this approach, hence its exclusion from this study, the use of error correction combined with dynamic time-steps may compensate for the additional work load.
- x. *Enhancements to the virtual heat facility to provide a more aesthetically accurate simulation.* The ability to produce ‘virtual heat’ within the system represents one of the greatest opportunities produced by the study. Future systems could allow for convection flow, a more accurate approach to thermal conductivity and closer analysis of material properties. It is felt that a dedicated virtual heat system would represent an exciting and fruitful future field of study.

- xi. *Enhanced virtual heater system.* This would operate with each spring calculating its distance from the virtual heat source before determining its exposure to radiation from the virtual heat source (see figure 7.1). The spring would analyse the levels of its own exposure relative to those of its neighbouring springs and perform a rationalisation with its adjoining connections.

Figure 7.1 – Virtual Heat Source Simulation Approach



7.5 Conclusions

The study has answered a number of questions concerning the suitability of three-dimensional mass-spring sheet structures. The results show that the approach does produce a viable solution to modelling sheet materials.

Three-dimensional sheet structures do provide an effective method of modelling sheet deformation, with the preliminary HCI study showing that aesthetic benefits can be gained with little or no degradation in performance. The most surprising aspect of this was the effectiveness of the hybrid 3D Provot mesh in outperforming the SLaMM and TriMM models without any substantial overhead in terms of speed.

The NURBS system has proven to be surprisingly effective in producing life-like simulations. The ability to smooth over imperfections in the mass-spring structures has produced a number of benefits to the visual appeal of the overall system although some enhancement to this basic approach taking into consideration collision detection issues would only further enhance this component, and ultimately the overall system.

Although only a limited investigation was conducted, the potential benefits of incorporating a virtual heat system into mass-spring systems were evident.

Of the MaSSE engines being employed, the second-order Euler-Richardson system has proven to be the most effective system (when taking into consideration all factors such as performance, time-step, system resources) although the fourth-order Runge-Kutta system would provide the most effective system (with minimal time performance penalty) should sufficient system resources be available. An investigation into the use of a fifth-Order Runge-Kutta MaSSE engine offers potential for future development, although it would require substantial efficiencies in relation to time-step size to negate the performance advantage demonstrated by the Euler-Richardson approach. The primary benefit of a fifth-order calculation engine that incorporates dynamic error checking would be to enhance the systems suitability for alternative forms of modelling, such as non-laminar three dimensional structures.

The potential to integrate the system with haptic technology provides an intriguing scope for future development of the system.

References

- Angel, E. (1999) *Interactive Computer Graphics – A Top Down Approach with OpenGL* (2nd Edition). Addison Wesley. ISBN: 0-201-38597-X. p539.
- Aono, M., Denti, P., Breen, D. and Wozny, M. (1996) *Fitting a Woven Cloth Model to a Curved Surface: Dart Insertion*. IEEE Computer Graphics and Applications, 1996 (September), pp 60-70.
- Aubel, A. and Thalmann, D. (2000) *Realistic Deformation of Human Body Shapes*. In Proceedings of Computer Animation and Simulation 2000, Interlaken, 2000 pp 125-135.
- Avril, Q., Gouranton, V., Arnaldi, B. (2009) New Trends in Collision Detection Performance. In Proceedings of VRIC, Lavel (France) Conference 2009.
- Bando, Yosuke; Chen, Bing-Yu; Nishita, Tomoyuki (2003) *Animating Hair with Loosely Connected Particles*. Computer Graphics Forum, Vol. 22 (3), September 2003, pp 411-418.
- Baraff, D. (1992) *Dynamic Simulation of Non-Penetrating Rigid Bodies*. PhD Thesis, Cornell University, NY, March 1992.
- Baraff, D. and Witkin, A. (1992) *Dynamic Simulation of Non-penetrating Flexible Bodies*. In Proceedings of Computer Graphics Annual Conference, pp 303-308.
- Baraff, D. and Witkin, A. (1998) *Large Steps in Cloth Simulation*. In Proceedings of Computer Graphics Annual Conference (SIGGRAPH 98), pp 43-54.
- Barzel, R. (1992) *Physically-Based Modeling for Computer Graphics*. Academic Press, 1992. ISBN: 0-12-079880-8.
- Barzel, R. (1997) *Faking Dynamics of Ropes and Springs*. IEEE Computer Graphics and Applications, 1997 (May-June), pp 31-39.

Beer, F., Johnston, E., Eisenberg, E., Clausen, W., Mazurek, D., Cornwell, P. (2007) *Vector Mechanics for Engineers: Statics and Dynamics (8th Edition)* McGraw Hill ISBN: 0072976985.

Bickerstaff, M. and Hellestrand, G., (1991) *A Highly Parallel Architecture for Real-Time Collision Detection in Flight Simulation*. Computers and Graphics, 1991. Vol. 15 (3), pp355-363.

Boissieux, L., Kiss, G., Magnenat –Thalmann, N., Kalra, P. (2000) *Simulation of Skin Aging and Wrinkles with Cosmetics Insight*, Computer Animation and Simulation, August 2000, pp15-27.

Boulic, R., Capin, T., Huang, Z., Kalra, P., Lintermann, B., Magnenat-Thalmann, N., Moccozet, L., Molet, T., Pandzic, I., Saar, K., Schmitt, A., Schen, J., and Thalmann, D. (1995) *The HUMANOID Environment for Interactive Animation of Multiple Deformable Human Characters*, Eurographics, Vol. 14 (3), pp 337-348.

Bourguignon, D. and Cani, M.P. (2000) *Controlling Anisotropy in Mass-Spring Systems*. In Proceedings of 11th Eurographics Workshop, Interlaken, Switzerland, pp 113-123.

Breen, D. (Organiser), Eischen, J., Kass, M., Magnenat-Thalmann, N., Vecchione, M. (Panelists). (1997) *Can We Get There from Here? : Current Challenges in Cloth Design, Modelling and Animation*. In Proceedings of Computer Graphics Annual Conference (SIGGRAPH 97). Los Angeles, California, USA, pp 437-439.

Bukowski, R. and C. Séquin (1997). *Interactive Simulation of Fire in Virtual Building Environments*. In Computer Graphics Annual Conference (SIGGRAPH 97), Los Angeles, California, USA, pp 35-44.

Chandrupatla, T. and Belegundu, A. (1997) *Introduction to Finite Elements in Engineering (2nd Edition)*. Prentice-Hall. 1997. ISBN 0-13-273319-6.

Chang, J., Zhang, J., Zia, R. (2009) *Modelling deformations in car crash animation*. In The Visual Computer, Vol. 25 (12), pp 1063 – 1072.

Cheng, B. and Topping, B. (1999) *Parallel Adaptive Mesh Generation and Geometric Modelling Using Nurbs*. Innovative Computational Methods for Structural Mechanics, pp 163-194.

Choi, K. (2006) *Interactive Cutting of Deformable Objects using Force Propagation Approach and Digital Design Analogy*. Computers & Graphics, Vol. 30 (2), April 2006, pp 233-243.

Choi, K. and Ko, H. (2005) *Research Problems in Clothing Simulation* Computer-Aided Design, Vol. 37, Issue 6, May 2005, pp 585-592.

Cordier, F. and Magnenat-Thalmann, N. (2005) *A Data-Driven Approach for Real-Time Clothes Simulation*, Proceedings of Computer Graphics Forum Vol.24(2), pp 173-183.

Cugini, U., Bordegoni, M., Rizzi, C., De Angelis, F. and Prati, M. (1999) *Modelling and Haptic Interaction with Non-Rigid Materials*. In Proceedings of Eurographics '99.

Dai, X., Furukawa, T., Mitsui, S. Takatera, M. and Shimizu, Y. (2001) *Drape Formation Based On Geometric Constraints and its Application to Skirt Modelling*. In International Journal of Clothing Science and Technology. Vol. 13 (1), 2001, pp 23-37.

Dai, X., Furukawa, T., Takatera, M., Hanazato, T. and Shimizu, Y. (1998) *Skirt Modeling Based On Paper Pattern and Its Application to Interactive Apparel Design and Manufacture*. In Proceedings of VSMM'98, November 1998, pp 104-109.

Debunne, G., Desbrun, M., Cani, M., Barr, A. (2000) *Adaptive Simulation of Soft Bodies in Real-Time*. In Proceedings of Computer Animation 2000, Philadelphia, USA, May 2000.

Debunne, G., Desbrun, M., Barr, A., Cani, M. (1999) *Interactive Multiresolution Animation of Deformable Models*. In Proceedings of 10th Eurographics workshop on Computer Animation and Simulation (CAS'99), September 1999.

Decaudin, P., Julius, D., Wither, J., Boissieux, L., Sheffer, A. and Cani, M. (2006) *Virtual Garments: A Fully Geometric Approach for Clothing Design*. In Proceedings of Eurographics 2006 Vol. 25(3), pp 625-634.

- DeRose, T., Kass, M. and Truong, T. (1998) *Subdivision Surfaces in Character Animation*. In Proceedings of ACM Computer Graphics Annual Conference (SIGGRAPH 98), pp 85-94.
- Desbrun, M. and Gascuel, M.. (1996) *Smoothed Particles: a New Paradigm for Animating Highly Deformable Bodies*. In Proceedings of 6th Eurographics Workshop on Animation and Simulation '96. 1996. Poitiers, France.
- Desbrun, M. and Gascuel, M. (1995) *Animating Soft Substances with Implicit Surfaces*, Proceedings of Computer Graphics Annual Conference, pp 287-290.
- Desbrun, M., Schroder, P. and Barr, A. (1999) *Interactive Animation of Structured Deformable Objects*. In Proceedings of Graphics Interface '99. pp 1-8.
- Eberhardt, B. and Weber, A. (1999). *A Particle System Approach to Knitted Textiles*. In Computers & Graphics. Vol. 23 (4), August, 1999, pp 599-606.
- Eberhardt, B., Weber, A., and Strasser, W. (1996) *A Fast, Flexible, Particle-System Model for Cloth Draping*. IEEE Computer Graphics and Applications, 1996 (September), pp 52-59.
- Filipczyk, P. and Nikiel, S. (2008) *Real-Time Simulation of a Sailboat*. In Proceedings of 2008 Conference on Human System Interactions, pp 970 – 974.
- Foley, J., van Dam, A., Feiner, S. and Hughes, J. (1990) *Computer Graphics: Principles and Practice*. The Systems Programming Series. Addison Wesley. ISBN: 0-201-12110-7.
- Fontana, M., Rizzi, C., and Cugini, U. (2006) *A CAD-Oriented Cloth Simulation System with Stable and Efficient ODE Solution*. Computers & Graphics. Vol. 30, pp 391-406.
- Fontana, M., Rizzi, C., and Cugini, U. (2005) *3D Virtual Apparel Design for Industrial Applications*. Computer-Aided Design. Vol. 37, pp 609-622.
- Gascuel, M. (1993) *An Implicit Formulation for Precise Contact Modelling between Flexible Solids*. In Proceedings of Computer Graphics Annual Conference, pp 313-320.
- Gould, H. and Tobochnik, J., (1996) *An Introduction to Computer Simulation Methods (2nd Edition)*, Addison Wesley. ISBN: 0-201-50604-1.

- Guan, Z., Ling, J., Tao, N., Ping, X. and Rongxi, T. (1997) *Study and Application of Physics-Based Deformable Curves and Surfaces*, Computers & Graphics. Vol. 21 (3) pp 305-313.
- Gudukbay, U., Ozguc, B., and Tokad, Y. (1997) *A Spring Force Formulation for Elastically Deformable Models*. Computers & Graphics, 1997. 21(3), pp 335-346.
- Haiyan, Y. and Zhaofeng, G. (2008) *An Improved Mass-Spring Model to Simulate Draping Cloth*. In Proceedings of 2008 International Conference on Intelligent Computation Technology and Automation, pp 568 – 571.
- Harada, T., Koshizuka, S. and Kawaguchi, Y. (2007) *Real-time Fluid Simulation Coupled with Cloth*. EG UK Theory and Practice of Computer Graphics.
- Higgins, R.A., (1994) *Properties of Engineering Materials (2nd Edition)*, Edward. ISBN 0-340-60033-0.
- Hinton, E. and Owen, D. (1979) *An Introduction to Finite Element Computations*. Pineridge Press Ltd. 1979. ISBN 0-906674-06-9.
- Howlett, P. and Hewitt, W. (1998) *Mass-Spring Simulation Using Adaptive Non-Active Points*. In Eurographics. Vol. 17(3). pp345-353.
- Hutchinson, D., Preston, M. and Hewitt, T. (1997) *Adaptive Refinement for Mass/Spring Simulations*. In 7th Eurographics Workshop on Animation and Simulation.
- James, D. and Pai, D. (1999) *Accurate Real-Time Deformable Objects*, Los Angeles, California, USA, August, 1999, pp 65-72.
- Jiang, Y. and Liu, Z. (2008) *Application and Research of Numerical Integration in Cloth Simulation*. In Proceedings of 2008 International Symposium on Computer Science and Computational Technology, pp128 – 132.
- Jiménez, P., Thomas, F., Torras, C. (2001) *3D Collision Detection: A Survey*. Computers & Graphics, Vol. 25, pp269-285.
- Joukhadar, A., Deguet, A., Laugier, C. (1997) *Towards Realistic Dynamic Simulation: Deformations and Collisions Models* In Proceedings of Workshop Dynamic Simulation: Methods and Applications p21-31, Grenoble, France, September 1997

- Kalra, P., Magnenat-Thalmann, N., Moccozet, L., Sannier, G., Aubel, A. and Thalmann, D. (1998) *Real-Time Animation of Realistic Virtual Humans*. IEEE Computer Graphics and Applications, Vol. 18 (5), pp 42-55.
- Kang, M. and Lee, J. (2007) *A Real-Time Cloth Draping Simulation Algorithm using Conjugate Harmonic Functions*. Computers & Graphics, Vol. 31 (2), April 2007, pp 271-279.
- Kergosien, Y., Gotoda, H., and Kunii, T., (1994) *Bending and Creasing Virtual Paper*. IEEE Computer Graphics & Applications, Jan., pp 40-48.
- Kilgard, M. (1994) *OpenGL and X, Column 1: An OpenGL Toolkit*. The X Journal. November/December 1994.
- Kim, D., Heo, J., Huh, J., Kim, J., Yoon, S. (2009) *HPCCD: Hybrid Parallel Continuous Collision Detection using CPUs and GPUs*. In Computer Graphics Forum, Vol. 28 (7), pp 1791-1800.
- Kim, T. and Venrovsky E. (2008) *DrivenShape – A Data-Driven Approach For Shape Deformation*. In Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp 49 - 55.
- Lamouret, A. and Gascuel, M. (1996) *Scripting Interactive Physically-based Motions with Relative Paths and Synchronization*. Computer Graphics Forum, Vol. 15 (1), pp25-34.
- Lamousin, H. and Waggenspack, W. (1994) *NURBS-Based Free-Form Deformations*. IEEE Computer Graphics and Applications, 1994 (November), pp 59-65.
- Lewis, J., Cordner, M. and Nickson, F. (2000) *Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation*. In Proceedings of Computer Graphics and Interactive Techniques (SIGGRAPH 2000), pp 165-172.
- Li, X. and Moshell, J. (1993) *Modelling Soil: Realtime Dynamic Models for Soil Slippage and Manipulation*. In Proceedings of Computer Graphics Annual Conference, pp 361-368.

- Lin, M. and Gottschalk, S. (1998) *Collision Detection between Geometric Models: A Survey*. In Proceedings of IMA Conference on Mathematics of Surfaces.
- Ling, L. (1997) *Animation of Stochastic Motion of 3D Cloth*. In Computers and Graphics, 1997. Vol. 21 (6), pp 769-775.
- Ling, L., Damodran, M., and Gay, R. (1996) *A Model for Animating the Motion of Cloth*. Computers and Graphics, 1996. Vol. 20 (1), pp 137-156.
- Liu, J., Ko, M. and Chang, R. (1998) *A Simple Self-Collision Avoidance for Cloth Animation*. In Computers and Graphics, Vol. 22 (1), pp 117-128.
- Liu, S., Liu, Q., An, T., Sun, J. (2009) *Physically Based Simulation of Thin Shell Objects' Burning*. In The Visual Computer, Vol. 25 (5-7), pp 687 – 696.
- Liu, Y., Zhang, D., Yuen, M. (2010) *A Survey on CAD Methods in 3D Garment Design*. In Computers in Industry.
- Luciani, A., Habibi, A., and Manzotti, E. (1995) *A Multi-Scale Physical Model of Granular Materials*. In Proceedings of Graphics Interface '95. 1995.
- Magenat-Thalmann, N. and Volino, P. (2005) *From Early Draping to Haute Couture Models: 20 Years of Research*. The Visual Computer, 2005. Vol. 21 (8), pp 506–519.
- Mahal, B.S., Clark, D.E.R., Simmons, J.E.L. and Shanks, G.C. (2004) *Mass-Spring Systems and Heat Simulation - A Framework for Virtual Manufacture*. In Proceedings Simulation Interoperability Standards Organisation (SISO) Fall 2004 Workshop. Orlando, USA, 19-24 September 2004.
- Mahal, B.S., Clark, D.E.R. and Simmons, J.E.L. (1999) *Investigation of Sheet Bending using Mass-Spring Systems within a Virtual Environment*. In Proceedings of 2nd International Conference/Workshop on Presence. Colchester, UK, April 1999.
- Maurel, W. and Thalmann, D. (2000) *Human Shoulder Modelling Including Scapulo-Thoracic Constraint and Joint Sinus Cones*. Computers & Graphics, Vol. 24 (2000), pp 203-218.
- Meng, Y., Mok, P., Jin, X. (2010) *Interactive Virtual Try-On Clothing Design Systems*. In Computer-Aided Design, Vol. 42 (4), pp 310-321.

- Meyer, M., DeBunne, G., Desbrun, M. and Barr, A. (2000) *Interactive Animation of Cloth-like Objects in Virtual Reality*. Journal of Visualization and Computer Animation 2000.
- Mirtich, B. (1996) *Impulse-based Dynamic Simulation of Rigid Body Systems*. Ph.D. thesis, University of California, Berkeley, December 1996.
- Montagnat, J., Delingette, H. and Ayache, N. (2001) *A Review of Deformable Surfaces: Topology, Geometry and Deformation*. In Image and Vision Computing, Vol. 19, pp 1023-1040.
- Nealen, A., Müller, M., Keiser, R., Boxerman, E., Carlson, M. (2006) *Physically Based Deformable Models in Computer Graphics*. Computer Graphics Forum, Vol. 25 (4), December 2006, pp 809-836.
- Nedel, L. and Thalmann, D. (1998) *Real Time Muscle Deformations Using Mass-Spring Systems*. In Proceedings of CGI'98, IEEE Computer Society Press.
- Neider, J., Davis, T. and Woo, M., *OpenGL Programming Guide*. 1993: Addison Wesley. ISBN: 0-201-63274-8.
- Nesme, M., Faure, F., Payan, Y. (2010) *Accurate Interactive Animation of Deformable Models at Arbitrary Resolution*. In International Journal of Image and Graphics, Vol. 10 (2), pp 1 -30.
- Ng, H., Grimsdale, R. and Allen, W. (1995) *A System for Modelling and Visualization of Cloth Material*. Computers and Graphics, 1995. Vol. 19 (3), pp 423-430.
- Ng, H. and Grimsdale, R., (1996) *Computer Graphics Techniques for Modeling Cloth*. IEEE Computer Graphics and Applications, 1996(September): p. 28-41.
- Onoue, K. and Nishita, T. (2005) *An Interactive Deformation System for Granular Material*. Computer Graphics Forum, Vol. 24 (1), March 2005, pp 51-60.
- Ortega, L., Feito, F., Grima, C. And Márquez, A. (2004) *Collision Detection using Polar Diagrams*. In Proceedings of Computer Graphics, Visualization and Computer Vision'2004 (WSCG'2004). pp117-121.

- Pabst S., Krzywinski, S., Schenk, A. and Thomaszewski, B. (2008) *Seams and Bending in Cloth Simulation*. In Proceedings of Workshop in Virtual Reality Interactions and Physical Simulation (VRIPHYS 2008).
- Pabst S., Thomaszewski, B., Strasser, W. (2009) *Anisotropic Friction for Deformable Surfaces and Solids*. In Proceedings of ACM SIGGRAPH Symposium on Computer Animation (VRIPHYS 2008), pp149 - 154.
- Pereira, E. and Moitinho de Almeida, J. (1999) Use of Hybrid Mixed Finite Element Models for the Characterisation of the Boundary Layer in Reissner-Mindlin Plates. *Innovative Computational Methods for Structural Mechanics*, p97.
- Plath, J. (2000) *Realistic Modelling of Textiles using Interacting Particle Systems*. In *Computers & Graphics*, 2000, Vol. 24 (6), pp 897-905.
- Preston, M. and Hewitt, W. (1994) *Animation using NURBS*. In *Computer Graphics Forum*. Vol. 13 (4). pp137-146.
- Provot, X. (1995) Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behaviour. In Proceedings of Graphics Interface '95, pp 147-154.
- Provot, X. (1997) Collision and Self-Collision Handling in Cloth Model Dedicated to Design Garments. In Proceedings of Graphics Interface '97. 1997.
- Rao, S., (1990) *Mechanical Vibrations (2nd Edition)*, Addison Wesley. ISBN: 0-201-52581-X.
- Rodriguez-Navarro, J. and Susin, A. (2006) *Non Structured Meshes for Cloth GPU Simulation Using FEM*. In Proceedings of Workshop on Virtual Reality Interaction and Physical Simulation 2006.
- Rodriguez-Navarro, J., Sainz, M., Susin, A. (2005) *GPU Based Cloth Simulation with Moving Humanoids*. In Proceeds of EUROGRAPHICS 2005
- Ryder, G. (1969) *Strength of Materials*, Palgrave Macmillan, ISBN 0333109287.
- Sayem, A., Kennon, R., Clarke, N. (2010) *3D CAD Systems for the Clothing Industry*. In *International Journal of Fashion Design, Technology and Education*, Vol. 3 (2), pp 45 – 53.

- Senin, M., Kojekine, N., Savchenko, V. and Hagiwara, I. (2003) *Particle-Based Collision Detection*. In Proceedings of Eurographics 2003.
- Shearer, M., Gremaud, P. and Kleiner, K. (2009) *Periodic Motion of a Mass–Spring System*. In IMA Journal of Applied Mathematics, Vol. 74 (6), pp 807-826.
- Silicon Graphics, I. (1996) *Direct 3D Analysis*. 1996, Silicon Graphics Incorporated.
- Smith, A., et al. (1995) A Simple and Efficient Method for Accurate Collision Detection Among Deformable Polyhedral Objects in Arbitrary Motion. In Proceedings of IEEE Annual Virtual Reality International Symposium. 1995.
- Song, M. and Grogono, P. (2009) *Application of Advanced Rendering and Animation Techniques for 3D Games to Softbody Modeling and Animation*. In Proceedings of Canadian Conference on Computer Science & Software Engineering, C3S2E 2009, pp 89 – 100.
- Steidel, R. (1989) *An Introduction to Mechanical Vibrations (3rd Edition)*. John Wiley and Sons. ISBN: 0-471-84545-0.
- Stewart, A. (1999) Computing *Visibility from Folded Surfaces*. In Computers and Graphics. Vol. 23 (1), pp 693-702.
- Sutherland, I., (1963) *SketchPad: A Man-Machine Graphical Communication System*. AFIPS Conference Proceedings 23, pp323-328.
- Sun, M. and Fiume, E. (1996) *A Technique for Constructing Developable Surfaces*. In Proceedings of Graphics Interface '96, pp 176-185.
- Sweeney, J., Caton-Rose, P. and Coates, P. (2002) *The Modelling of Large Deformations of Pre-Oriented Polyethylene*. In Polymer, Vol. 43 (2002), pp 899-907.
- Tang, M., Manocha, D., Tong, R. (2009) *Multi-core Collision Detection Between Deformable Models*. In Proceedings of 2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling, pp 355-360.
- Terasawa, M. (1996) *Physically-Based Computer Animation Controlled by Motion Features*. In International Journal of Shape Modeling. Vol.2 (4), pp 247-256.

Teschner, M., Kimmerle, S., Heidelberger, B., Zachmann, G., Raghupathi, L., Fuhrmann, A., Cani, M-P., Faure, F., Magnenat-Thalmann, N., Strasser, W. and Volino, P. (2005) *Collision Detection for Deformable Objects*. In Proceedings of Computer Graphics Forum. Vol. 24 (1), pp 61-81.

Theisel, H. and Kreuseler, M. (1998) *An Enhanced Spring Model for Information Visualization*. In Proceedings of Eurographics. Vol. 17 (3), pp 334-344.

Turner, R. and Gobbetti, E. (1998) *Interactive Construction and Animation of Layered Elastically Deformable Characters*. In Computer Graphics Forum, Vol. 17 (2), pp 135-152.

Vassilev, T., Spanlang, B. and Chrysanthou, Y. (2001) *Efficient Cloth Model and Collisions Detection for Dressing Virtual People*. In Proceedings of ACM/EG Games Technology Conference, January, 2001.

Vemuri, B., Cao, Y. and Chen, L. (1998) *Fast Collision Detection Algorithms with Applications to Particle Flow*. In Eurographics. Vol. 17 (3), pp 121-134.

Vince, J. (2000) *Essential Computer Animation Fast*. Springer-Verlag. ISBN 1-85233-141-0.

Volino, P. (2005) *Collision Detection for Deformable Objects*. In Proceedings of Computer Graphics Forum. Vol. 24 (1), pp 61-81.

Volino, P., Cordier, F. and Magnenat-Thalmann, N. (2005) *From Early Virtual Garment Simulation to Interactive Fashion Design*. In Computer-Aided Design Journal, Vol. 37, pp 593 – 608.

Volino, P.; Davy, P.; Bonanni, U., Luible, C., Magnenat-Thalmann, N., Mäkinen, M., Meinander, H. (2007) *From Measured Physical Parameters to the Haptic Feeling of Fabric*. In The Visual Computer, Vol. 23 (2), pp 133 – 142.

Volino, P. and Magnenat-Thalmann, N. (2006) *Simple Linear Bending Stiffness in particle Systems*. In Proceedings of 2006 ACM SIGGRAPH Eurographics Symposium on Computer Animation, pp101 - 105.

Volino, P. and Magnenat-Thalmann, N. (2000a) *Virtual Clothing Theory and Practice*. Springer, ISBN 783540676003.

Volino, P. and Magnenat-Thalmann, N. (2000b) *Accurate Collision Response on Polygonal Meshes*. In Proceedings of Computer Animation 2000, Annual Conference Series, IEEE Press, May 2000.

Volino, P. and Magnenat-Thalmann, N. (1994) Efficient Self-Collision Detection on Smoothly Discretized Surface Animations using Geometrical Shape. Eurographics, Vol. 13 (3), pp 155-166.

Volino, P., Magnenat-Thalmann, N. and Faure, F. (2009) *A Simple Approach to Nonlinear Tensile Stiffness for Accurate Cloth Simulation*. In ACM Transaction on Graphics, Vol. 28 (4).

Wallner, G. (2007) *Simulating, Animating and Rendering Clothes*. In Seminare aus Computergraphik im Sommersemester (Technische Universität Wien) 2007.

Wilhelms, J. (1997) *Animals with Anatomy*. In IEEE Computer Graphics and Applications, 1997 (May-June), pp 22-30.

Wilhelms, J. and Van Gelder, A. (1997) *Anatomically Based Modelling*. In Proceedings of Computer Graphics Annual Conference (SIGGRAPH97), August, 1997, pp 173-180.

Wilson, A., Larsen, E., Manocha, D. and Lin, M. (1999) *Partitioning and Handling Massive Models for Interactive Collision Detection*. In Proceedings of Eurographics. Vol. 18(3), pp 319-329.

Witting, P. (1999) *Computational Fluid Dynamics in a Traditional Animation Environment*. In Proceedings of Computer Graphics Conference Series (SIGGRAPH '99), Los Angeles, USA, pp 129-136.

Wu, Z., Au, C., Yuen, M. (2003) *Mechanical Properties of Fabric Materials for Draping*. In International Journal of Clothing Science and Technology, Vol. 15 (1), pp 56-68.

Zhang, D. and Yuen, M. (2001) *Cloth Simulation Using Multilevel Meshes*. In Computers & Graphics. Vol. 25 (3), June, 2001, pp 383-389.

Zhang, D. and Yuen, M. (2002) *A Coherence-Based Collision Detection Method for Dressed Human Simulation*. Computer Graphics Forum, Vol. 21(1), pp 33-42.

Zhong, Y. (2010) *Redressing Three-dimensional Garments Based on Pose Duplication*. In Textile Research Journal, Vol. 80 (10), pp 904-916.

Zhong, Y. (2009) *Fast Penetration Resolving for Multi-layered Virtual Garment Dressing*. In Textile Research Journal, Vol. 79 (10), pp 815-821.

Zhong, Y., Shirinzadeh, B., Smith, B., Gu, C. (2009) *An Electromechanical Based Deformable Model for Soft Tissue Simulation*. In Artificial Intelligence in Medicine, Vol. 47 (3), pp 275-288.

Zhong, Y. and Xu, B. (2009) *Three-dimensional Garment Dressing Simulation*. In Textile Research Journal, Vol. 79 (10), pp 792 - 803.

Zhongke, W., Feng, L. and Hock, S. (2003) *Topology Preserving Voxelisation of Rational Bézier and NURBS Curves*. Computing & Graphics. Vol. 27(1), pp 83-89.

8 Appendices

8.1 Appendix A - Application Development

8.1.1 Tool Selection

To generate the simulated sheet application an appropriate tool and/or development environment must first be created and evaluated. These tools and equipment must, as a minimum, include a hardware platform, a software compiler, and/or any graphics libraries and possibly a graphics modeller. This section will then examine each aspect in detail.

8.1.2 Hardware Selection

Due to the fast pace of change within immersive technology aspects of Virtual Environments, the decision was made early in the project lifecycle to implement the final system within a non-immersive, desktop-based environment. This way the final system would be less hardware dependent and capable of adopting the latest technology available at the time, rather than being locked into an older approach that may be outdated prior to the completion of the final product.

A number of different hardware platforms exist upon which the sheet simulation may be developed. Factors for the choice include:

- The utilisation of an inexpensive system that does not require highly specialised or non-generic equipment, as stipulated in the thesis hypothesis.
- No use of dedicated technologies that may allow the simulation application to become prematurely defunct due to reliance upon an outdated technology.
- A reliable and stable environment that will provided a suitable backdrop, not only for development but also use of the simulation.

It was for these reasons a Personal Computer⁴ (PC) based hardware choice was made. The PC is the most ubiquitous computing hardware platform in the world today, offering an inexpensive⁵ environment with unprecedented performance enhancements. A second advantage of PC equipment choice was the dominance of two compatible operating systems on the platform, that of Microsoft Windows 98/ME and Windows 2000/XP. Arguably there are several equally suitable Operating Systems (OS) available on the Intel platform, such as the PC derivatives of the UNIX OS (LINUX). However none currently offer the level of support or flexibility offered by Windows XP software development tools such as Visual C++ and Microsoft Foundation Classes (MFC). In addition, the dominance of the Windows platform ensures that the final system will be executable upon any Windows computer. Also the use of Commercial Off-The-Shelf (COTS) technology has recently started to gain significant momentum within areas of industry traditionally associated with expensive, bespoke systems (e.g. Defence, Manufacturing). The benefits of economies of scale have helped to encourage rapidly advancing, generic technology at substantially reduced costs. This has resulted in many commercial and industrial corporations adopting a similar approach to that presented here for their own product development.

8.1.3 Development Environment

There are two principal choices that must be made at the start of any graphics-based application; which language and which (if any) graphics libraries and toolkits are to be employed to deal with windowing, volume rendering and other such issues.

The term ‘Object Oriented’ (OO) has become synonymous with the creation of well-structured, modular and scaleable computer code. It was decided early in the project lifecycle that an Object Oriented approach should be adopted, due to the advantages that such a methodology could provide in application flexibility.

Two principal OO languages offer themselves to the project, each carrying their own advantageous characteristics, these being Smalltalk and C++. Due to the rapid acceptance of C++ (and its subsequent PC-based evolution, Microsoft’s Visual C++) and decline of Smalltalk, Microsoft’s own Visual C++ Developer toolkit was selected.

⁴ PC denotes a 100% IBM compatible Personal Computer

⁵ CPU and graphics based price to performance ratio

The second stage in the selection of a development environment is the choice of Graphics Library (GL). A graphics library is a collection of pre-written subroutines that perform low-level graphics related tasks. Two industry standard graphics systems exist for the PC platform, OpenGL and DirectX.

The Open Graphics Library (OpenGL) consists of a set of interchangeable graphics commands that are available across all of the major workstation and personal computer platforms. This is in stark contrast to DirectX which, as a proprietary standard, is governed solely by the Microsoft Corporation. Following an analysis of the two libraries (which are summarised in Table A.1) the decision was made to adopt OpenGL as the library of choice.

<i>Table A.1 - Overview of Direct3D/DirectX and OpenGL.</i>		
	DirectX	OpenGL
Target Market	Games	Generic (i.e. Animation, Games, Modelling & Simulation)
Platform	DOS, Microsoft Windows (XP, Vista, 7)	UNIX (all major derivatives), MS Windows (all current releases)
Current Release	11	4.1
Standards Body	Microsoft Corporation	Architecture Review Board
Command Level	Mid-level	Low/Mid-level
Cost	Free	Free (PC & most UNIX Platforms)
H/W Support	Good ⁶	Good ³

To resolve the issue of the low-level nature of OpenGL the decision was made to utilise the Graphics Library Utility Toolkit (GLuT), a window system independent toolkit for

writing OpenGL programs that implements a simple windowing application programming interface (API) for OpenGL (Silicon Graphics, 1999). The advantages of using GLUT is that amongst additional benefits it provides a host of services that facilitate the use of input devices, providing an operating system-independent environment for generating windows and assisting in the production of solid and wire-frame models. Although GLUT offers substantial gains over the standard OpenGL library both it, and OpenGL have not yet been adequately established to cater for the advanced windowing techniques required by modern applications. Therefore, to allow fully interactive user input into the system during runtime, the GLUi (Graphics Library User Interface) was merged into the system.

⁶ Dependent upon graphics card/hardware used in the PC

8.2 Appendix B – Principles of Physical Simulation

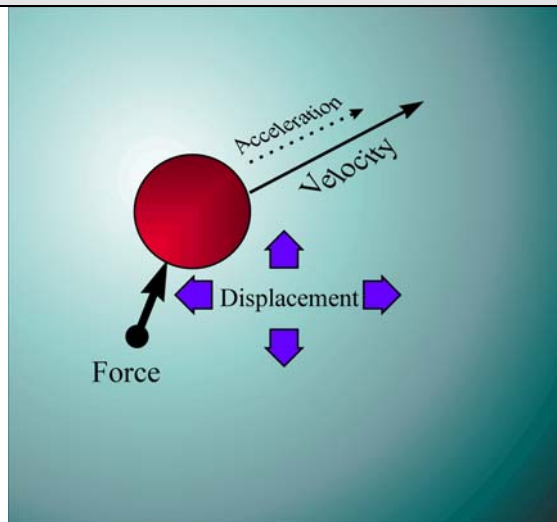
8.2.1 Mass Systems

Each *mass* instance (or sheet control point), when acting as an individual particle in free space obeys Newton's second law:

$$ma = f \quad (\text{Equation 8.1})$$

Where the mass m of the particle multiplied by the particle's acceleration a is equal to the sum of the forces f acting upon it. Based upon this expression, a position and velocity matrix (or vector) is used to determine force and acceleration. In a free-space system of n independent particles, the force calculation per control point is of order $O(n)$. When this is adapted to take into account interactions with other particles it results in a force calculation of order $O(n^2)$. In conventional particle systems, this figure would be reduced to take into account the effects of only those particles that are close to it.

Figure B.1 - A Single Particle



8.2.2 Elastic Collisions in Physical Simulation

Elastic collisions may involve a wide variety of factors that would influence the outcome of the clash, such as:

- Irregular boundaries between the objects.
- Different friction coefficients between the two surfaces.
- Non-homogeneous surface densities.
- The objects may possess some form of angular momentum.
- The objects might be travelling across an irregular surface that itself introduces non-linear frictional surfaces.

However, since the project would only involve ‘simplistic’ levels of collisions between the sheet material and the surface of either the floor or wall, then only a limited scope for elastic collisions will be investigated.

The conservation of momentum equation states that when given two particles m_a , m_b , and their respective velocities v_a , and v_b :

$$m_a v_a + m_b v_b = \text{constant} \quad (\text{Equation 8.2})$$

Meanwhile, the principle of relative motion is based upon the idea that after an impact where e represents the coefficient of restitution, the velocities of the two particles (u_a and u_b respectively) would be:

$$u_a - u_b = -e(v_a - v_b) \quad (\text{Equation 8.3})$$

Collision Detection and Response

This section deals with how an object would react when it has made contact with the ground, in particular how it would bounce off the surface.

The method described uses an impulse to change the velocity of masses to represent collision response. An impulse is essentially a large force that is integrated over a short period of time. Since our study is examining masses that, in essence, individually represent single particles in space, there is no need to examine the field of rigid body dynamics which are used to simulate the interaction between solid objects, such as two cubes colliding with each other.

Collision Detection

During the process of detecting collisions, a test value is first calculated (see figure B.2) using equation (8.4):

$$\text{Test value} = (X - P) \cdot N \quad (\text{Equation 8.4})$$

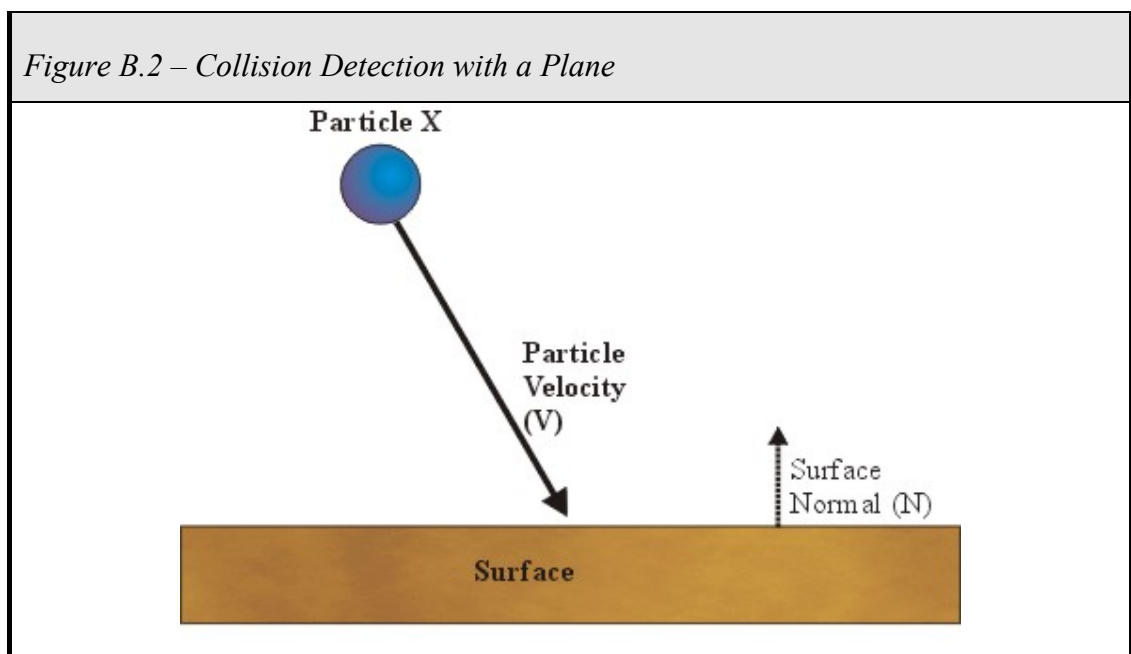
where

N = Normal vector of plane

V = Velocity of mass

P = Point on plane

e = <small threshold near 0>



Then based upon the value of the result of equation (8.4), it is possible to determine whether a collision has or is occurring using the criteria that:

- If test value $< e$ then some collision has occurred.
- If test value $= -e$ then the mass has passed through the plane
- If test value $= 0$ then the mass is in contact with the plane, therefore a further check is needed to determine whether the mass is moving closer or further away from the plane. This is achieved by checking if $N \cdot V < 0$. If this is the case then the mass and plane are in collision, otherwise the mass is moving further away from the wall.

Collision Response

Once it has been determined that a collision has taken place, then an appropriate response must be calculated to retain the realism of the system. This reaction must be based upon the angle, velocity and mass of the particle as it strikes the surface plane (see figure B.3).

The level of reaction is based upon the equations

$$V_n = (N \cdot V)N \quad (\text{Equation 8.5})$$

$$V_t = V - V_n \quad (\text{Equation 8.6})$$

$$V' = V_t - K_r V_n \quad (\text{Equation 8.7})$$

where

N = Normal (unit) vector of plane

P = Point on plane

X = Test point (aka the position of the mass)

V_n = Velocity normal (of mass to plane)

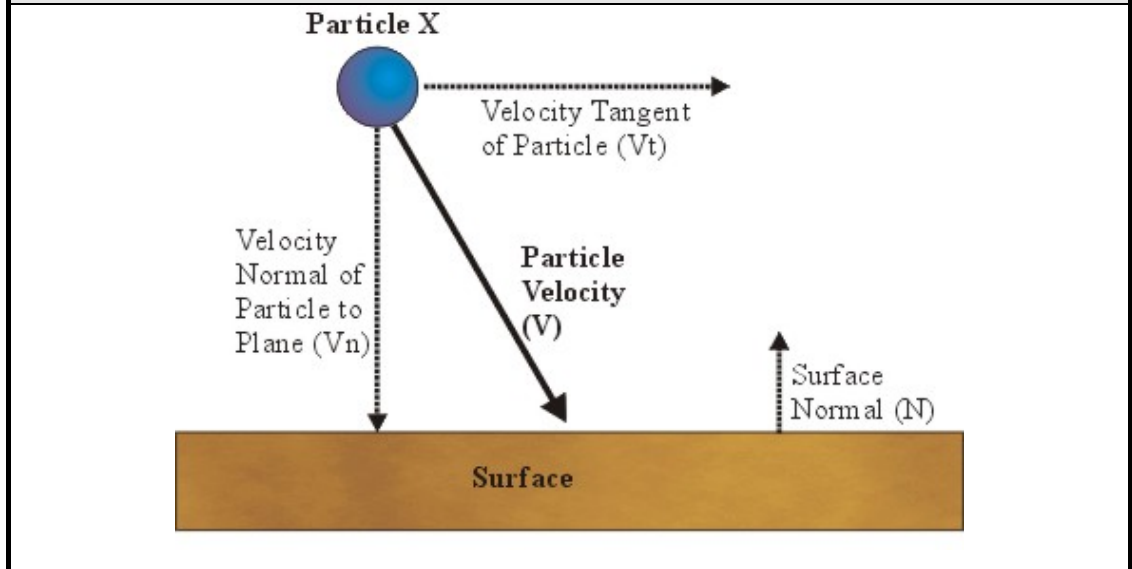
V = Velocity (of mass)

V_t = Velocity tangent (of mass)

V' = New velocity (after collision)

K_r = Restitution coefficient (with 0.0 = perfect plasticity and 1.0 = perfect elasticity)

Figure B.3 – Collision Response with a Plane



To ensure that Normal vector of the plane is a *unit* vector, it must be normalised so that:

$$\sqrt{x^2 + y^2 + z^2} = 1 \quad (\text{Equation 8.8})$$

Therefore the Normalised Unit Vector is defined as:

$$NUvec = \frac{Nvec}{\sqrt{Nx^2 + Ny^2 + Nz^2}} \quad (\text{Equation 8.9})$$

where

NU_{vec} = Each vector component of Normal Unit Vector

N_{vec} = Each vector component of Normal Vector (of the plane)

8.3 Appendix C – HCI Testing

8.3.1 Evaluation Procedure Sheet

Pre-Testing Talk.

Thank the user for participating in the test.

Explain that the test will take approximately 30 minutes.

The system to be evaluated will examine the visual accuracy and speed performance of four simulation programs compared to real-world photo-footage.

Explain what is to be asked of the test subject.

They will be asked to view the real-world video footage of the sheet falling onto an object. They will then be shown a simulation portraying the same scene.

They will be asked to complete part 1 of the user evaluation questionnaire.

Steps 1.4.1 and 1.4.2 will be repeated for the remaining 3 simulations.

The user will be asked to complete part 2 of the user evaluation questionnaire.

State that the system is being tested not them.

Mention that they may leave the test at any time and if they wish, the test results will not be used.

Hand Out General Computing Questionnaire

Perform Evaluation Tests (as described in 1.4.1, 1.4.2 and 1.4.3)

Complete User Evaluation Questionnaire Part 2.

Thank the user for their participation

General discussion period

Did they find the simulations provided a fair representation?

Were any areas of improvement identified?

Note down any additional comments

Thank the user for their participation

8.3.2 Background Questionnaire

ID Number: _____

Age: _____

Sex : Male / Female

How many years of computing experience do you have? _____

On average, how many HOURS do you spend on a computer each week?

☐ Less than 1 hour

☐ Between 1 and 4 hours

☐ Between 4 and 10 hours

☐ Between 10 and 20 hours

☐ Over 20 hours

Are you familiar with computer based simulation or virtual environments (excluding computer games)?

Yes / No

Are you familiar with advanced (A-Level or above) Physics or Engineering based principles?

Yes / No

8.3.3 User Satisfaction Questionnaire Part 1

This is a user evaluation questionnaire that is used for the purposes of data gathering during the evaluation of a real-time computer based simulation of modelling sheet deformation.

You're Rights:

You are not obliged to complete the questionnaire and if you do wish, these results will not be shown in any statistics.

You may withdraw from the test at any stage.

System Accuracy

A1. Sheet's Initial Deformation Reaction

Poor

Good

1 2 3 4 5 6 7 8 9 N/A

A2. Stretching Performance

Inadequate

Effective

1 2 3 4 5 6 7 8 9 N/A

A3. Sheet's Reforming Accuracy

Meagre

Successful

1 2 3 4 5 6 7 8 9 N/A

A4. Shape Retention of Sheet

Poor

Good

1 2 3 4 5 6 7 8 9 N/A

A5. Sheet's Sliding Performance

Inaccurate

Realistic

1 2 3 4 5 6 7 8 9 N/A

System Speed Performance

B1. Sheet's Speed Performance While Deforming

Slow

Fast

1 2 3 4 5 6 7 8 9 N/A

B2. Users Reaction to Performance

Frustrating

Satisfying

1 2 3 4 5 6 7 8 9 N/A

B3. Sheet's Overall Accuracy

Inaccurate

Accurate

1 2 3 4 5 6 7 8 9 N/A

8.3.4 User Satisfaction Questionnaire Part 2

This is a user evaluation questionnaire that is used for the purposes of data gathering during the evaluation of a real-time computer based simulation of modelling sheet deformation.

You're Rights:

You are not obliged to complete the questionnaire and if you do wish, these results will not be shown in any statistics.

You may withdraw from the test at any stage.

Overall User Reactions

C1. Overall Feeling of the Simulations compared to Real Footage

Negative

Positive

1 2 3 4 5 6 7 8 9 N/A

C2. Overall Feeling of the Visual Representation of the Sheet Simulations

Poor

Good

1 2 3 4 5 6 7 8 9 N/A

8.4 Appendix D – HCI Experimental Data

Category	Count	Percentage
Male	10	10.0%
Female	90	90.0%
Total	100	100.0%

[illegible][illegible]

Item No.	Score 0-10										Score 0-10										Score 0-10										Total									
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8
1	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
2	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
3	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
4	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
5	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
6	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
7	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
8	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
10	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
11	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
12	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
13	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
14	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
15	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
16	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
17	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
18	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
19	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
20	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
21	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
22	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
23	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
24	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
25	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
26	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
27	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
28	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
29	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
30	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
31	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
32	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
33	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
34	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
35	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
36	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
37	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
38	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
39	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
40	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
41	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
42	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
43	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
44	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
45	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
46	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
47	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
48	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
49	0	1	2	3</																																				

[illegible]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
4	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
5	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80																				

	SLaMM Sheet																											
	Cube (1)									Wedge #1 (4)									Wedge #2 (7)									
UserID	A1	A2	A3	A4	A5	B1	B2	B3	Total	A1	A2	A3	A4	A5	B1	B2	B3	Total	A1	A2	A3	A4	A5	B1	B2	B3	Total	
001	4	5	1	1	N/A	2	3	3	19	3	5	3	5	N/A	3	4	4	27	3	3	3	3	5	6	4	5	32	
002	3	5	2	2	4	3	2	3	24	7	7	8	8	8	3	8	8	57	2	5	3	3	5	7	3	3	31	
003	7	5	3	4	N/A	5	5	4	33	7	5	7	7	N/A	5	6	7	44	7	7	7	7	5	7	6	7	53	
004	8	7	6	7	N/A	8	5	5	46	5	5	6	6	N/A	6	5	5	38	7	7	7	7	8	7	8	7	58	
005	7	6	2	1	N/A	1	3	3	23	7	6	6	6	N/A	4	6	6	41	8	7	5	5	8	7	7	6	53	
006	6	8	4	5	7	4	6	6	46	4	8	5	3	7	7	7	4	45	8	8	8	8	8	9	8	6	63	
007	7	7	6	5	N/A	5	5	5	40	6	8	6	7	N/A	6	5	5	43	6	5	4	5	7	6	6	5	44	
008	6	7	6	7	N/A	6	6	6	44	8	8	7	7	N/A	7	8	8	53	7	8	6	6	7	6	7	6	53	
009	3	4	1	3	N/A	3	2	2	18	5	5	5	4	N/A	5	4	4	32	6	6	5	5	5	6	5	5	43	
010	7	8	7	6	7	5	5	7	52	9	9	9	9	9	8	9	9	71	9	9	9	9	9	9	9	9	72	
Total	58	62	38	41	18	42	42	44	345	61	66	62	62	24	54	62	60	451	63	65	57	58	67	70	63	59	502	
Mean	5.8	6.2	3.8	4.1	6	4.2	4.2	4.4	34.5	6.1	6.6	6.2	6.2	8	5.4	6.2	6	45.1	6.3	6.5	5.7	5.8	6.7	7	6.3	5.9	50.2	
Standard	1.72047	1.32665	2.18174	2.16564	1.41421	1.93907	1.46969	1.56205	12.0354	1.75784	1.49666	1.6	1.72047	0.8165	1.62481	1.66132	1.78885	12.0287	2.1	1.68819	1.95192	1.8868	1.48661	1.09545	1.79165	1.51327	12.3028	
Median	6.5	6.5	3.5	4.5	7	4.5	5	4.5	36.5	6.5	6.5	6	6.5	8	5.5	6	5.5	43.5	7	7	5.5	5.5	7	7	6.5	6	53	

Overall Simulation			
UserID	C1	C2	Total
001	4	4	55
002	4	5	53
003	6	6	85
004	7	7	94
005	7	3	83
006	7	7	100
007	7	7	75
008	6	8	86
009	4	3	66
010	8	8	115
Total	60	58	812
Mean	6	5.8	81.2
Standard	1.41421	1.83303	18.5785

	TRIMM Sheet																											
	Cube (2)										Wedge #1 (5)										Wedge #2 (8)							
UserID	A1	A2	A3	A4	A5	B1	B2	B3	Total	A1	A2	A3	A4	A5	B1	B2	B3	Total	A1	A2	A3	A4	A5	B1	B2	B3	Total	
001	4	5	3	3	N/A	3	4	4	26	3	3	2	2	N/A	2	2	2	16	3	3	2	2	3	3	3	4	23	
002	3	5	3	3	5	3	4	3	29	5	6	6	6	6	6	5	6	6	46	4	6	5	5	6	6	5	4	41
003	7	6	6	5	N/A	4	5	6	39	6	4	5	5	N/A	4	5	5	34	7	6	5	5	6	6	5	6	46	
004	8	7	4	7	N/A	8	4	4	42	5	5	6	6	N/A	7	6	5	40	8	8	6	7	8	8	7	7	59	
005	7	6	3	3	N/A	3	3	3	28	6	5	5	5	N/A	5	5	4	35	8	8	8	8	9	8	8	8	65	
006	8	8	6	6	8	6	5	6	53	7	6	5	4	6	6	4	6	44	8	8	8	6	4	8	6	7	6	53
007	7	6	5	6	N/A	6	5	5	40	6	6	5	6	N/A	6	5	6	40	6	7	7	6	7	7	6	7	52	
008	7	7	6	6	N/A	6	7	7	46	7	7	6	6	N/A	7	7	7	47	7	6	6	6	7	7	7	7	56	
009	3	4	2	2	N/A	5	2	2	20	5	5	5	4	N/A	5	4	4	32	6	6	6	6	7	7	6	6	5	49
010	8	8	7	7	8	7	6	7	58	8	9	8	8	9	9	9	9	69	8	8	8	8	8	8	8	8	65	
Total	62	62	45	48	21	51	45	47	381	58	56	53	52	21	56	53	54	403	65	66	61	58	70	65	62	62	509	
Mean	6.2	6.2	4.5	4.8	7	5.1	4.5	4.7	38.1	5.8	5.6	5.3	5.2	7	5.6	5.3	5.4	40.3	6.5	6.6	6.1	5.8	7	6.5	6.2	6.2	50.9	
Standard	1.93907	1.249	1.62788	1.77764	1.41421	1.7	1.36015	1.67631	11.6572	1.32665	1.56205	1.41774	1.53623	1.41421	1.8	1.79165	1.8	12.8144	1.68819	1.49666	1.75784	1.77764	1.67332	1.43178	1.46969	1.4	11.8275	
Median	7	6	4.5	5.5	8	5.5	4.5	4.5	39.5	6	5.5	5	5.5	6	5.5	5	5.5	40	7	6.5	6	6	7	6.5	6.5	6.5	52.5	

	Provot 3D Sheet																											
	Cube (3)										Wedge #1 (6)										Wedge #2 (9)							
UserID	A1	A2	A3	A4	A5	B1	B2	B3	Total	A1	A2	A3	A4	A5	B1	B2	B3	Total	A1	A2	A3	A4	A5	B1	B2	B3	Total	
001	4	4	5	5	N/A	3	4	5	30	3	3	2	2	N/A	3	2	2	17	5	4	8	7	5	5	5	5	44	
002	4	5	5	5	6	3	5	5	38	7	7	8	8	7	3	8	8	56	6	7	7	7	8	9	7	8	59	
003	7	6	6	6	N/A	6	5	7	43	7	6	6	5	N/A	5	5	6	40	7	7	7	7	7	6	6	7	53	
004	4	5	4	5	N/A	4	3	3	28	8	7	7	8	N/A	7	7	6	50	8	7	5	7	8	8	7	7	57	
005	7	4	4	4	N/A	3	4	4	30	7	7	5	6	N/A	5	5	5	40	8	8	9	9	6	7	8	7	62	
006	8	8	6	5	8	6	5	7	53	7	6	5	5	6	8	6	4	47	7	8	8	8	9	8	9	8	66	
007	7	6	8	6	N/A	7	7	6	47	7	6	7	6	N/A	7	7	7	47	6	6	6	6	7	7	7	6	52	
008	7	7	7	8	N/A	7	7	7	50	7	7	7	7	6	N/A	7	6	7	47	6	6	6	6	7	6	6	49	
009	4	4	2	2	N/A	2	2	2	18	5	5	7	6	N/A	6	5	6	40	6	7	7	7	6	6	4	4	46	
010	8	8	8	8	8	8	8	8	64	7	7	8	7	7	7	7	8	58	9	8	7	7	7	9	9	8	64	
Total	60	57	55	54	22	49	50	54	401	65	61	62	59	20	58	58	59	442	68	68	70	73	70	69	69	65	552	
Mean	6	5.7	5.5	5.4	7.33333	4.9	5	5.4	40.1	6.5	6.1	6.2	5.9	6.66667	5.8	5.8	5.9	44.2	6.8	6.8	7	7.3	7	6.9	6.9	6.5	55.2	
Standard	1.67332	1.48661	1.80278	1.68523	0.94281	2.02237	1.78885	1.85472	13.1716	1.36015	1.22066	1.72047	1.64012	0.4714	1.66132	1.6	1.75784	10.8609	1.16619	1.16619	1.09545	1.00499	1.26491	1.13578	1.57797	1.20416	7.22219	
Median	7	5.5	5.5	5	8	5	5	5.5	40.5	7	6.5	7	6	7	6.5	6	6	47	6.5	7	7	7	7	7	7	7	55	

	Provot 2D Sheet																											
	Cube										Wedge #1								Wedge #2									
UserID	A1	A2	A3	A4	A5	B1	B2	B3	Total	A1	A2	A3	A4	A5	B1	B2	B3	Total	A1	A2	A3	A4	A5	B1	B2	B3	Total	
001	4	4	4	4	N/A	3	4	4	27	3	3	2	2	N/A	3	2	2	17	4	3	3	3	3	3	4	4	4	28
002	3	5	3	3	5	3	3	3	28	6	6	6	8	6	3	7	6	48	3	6	5	4	5	6	5	4	38	
003	7	6	5	7	N/A	6	7	7	45	7	8	8	8	N/A	6	7	8	52	8	7	6	7	5	7	8	8	56	
004	8	7	6	6	N/A	7	6	7	47	7	5	6	6	N/A	7	6	6	43	7	6	7	7	6	8	7	7	55	
005	7	6	3	3	N/A	5	2	3	29	7	7	8	8	N/A	7	8	8	53	7	7	5	5	4	6	6	4	44	
006	8	8	7	8	7	6	7	7	58	8	7	8	8	7	9	9	9	65	9	8	7	6	7	8	7	6	58	
007	7	6	5	7	N/A	5	7	6	43	8	7	7	7	N/A	7	7	7	50	7	6	7	6	6	5	5	6	48	
008	6	6	5	5	N/A	6	5	5	38	7	7	8	7	N/A	7	7	7	50	7	7	5	5	5	6	6	6	47	
009	4	4	2	2	N/A	3	2	2	19	6	6	7	6	N/A	5	5	5	40	6	6	5	7	3	5	3	3	38	
010	7	7	6	7	7	8	7	6	55	6	6	6	6	7	6	7	6	51	7	7	6	6	6	6	7	6	52	
Total	61	59	46	52	19	52	50	50	389	65	62	66	67	19	61	65	64	469	65	63	56	56	50	62	58	54	464	
Mean	6.1	5.9	4.6	5.2	6.33333	5.2	5	5	38.9	6.5	6.2	6.6	6.7	6.33333	6.1	6.5	6.4	46.9	6.5	6.3	5.6	5.6	5	6.2	5.8	5.4	46.4	
Standard	1.7	1.22066	1.49666	1.98997	0.94281	1.66132	2	1.78885	12.243	1.36015	1.32665	1.74356	1.73494	0.4714	1.81384	1.80278	1.85472	11.7682	1.68819	1.26886	1.2	1.28062	1.26491	1.249	1.46969	1.49666	9.05759	
Median	7	6	5	5.5	7	5.5	5.5	5.5	40.5	7	6.5	7	7	6	7	7	6.5	50	7	6.5	5.5	6	5	6	6	6	47.5	